Sascha Kersken

Handbuch für Fachinformatiker

Der Ausbildungsbegleiter

Galileo Computing

Auf einen Blick

	Vorwort	17
1	Einführung	25
2	Mathematische und technische Grundlagen	53
3	Hardware	95
4	Betriebssysteme	145
5	Grundlagen der Programmierung	227
6	Konzepte der Programmierung	277
7	Software-Engineering	325
8	Datenbanken	355
9	Bildbearbeitung und Grafik	387
10	DTP und digitale Druckvorstufe	421
11	Multimedia	451
12	Datei- und Datenformate	515
13	Netzwerktechnik	537
14	Netzwerkanwendungen	635
15	XML	685
16	HTML, CSS und Webdesign	719
17	Webanwendungen	807
18	JavaScript	855
19	Flash MX 2004	915
20	Computer- und Netzwerksicherheit	971
A	Glossar	991
B	Kommentiertes Literaturverzeichnis	1005
	Index	1019

Inhalt

Vorwort 17

Kompendium der Informationstechnik – The Next Generation ... 17
Worum geht es in diesem Buch? ... 18
Für wen ist dieses Buch geeignet? ... 22
Danksagungen ... 23

1 Einführung 25

1.1 Informationstechnik, Informatik und EDV ... 25
1.1.1 Fachrichtungen der Informatik ... 25
1.1.2 Überblick über die IT-Ausbildung ... 26
1.2 Die Geschichte der Rechenmaschinen und Computer ... 32
1.2.1 Die Vorgeschichte ... 34
1.2.2 Die Entwicklung der elektronischen Rechner ... 35
1.2.3 Entwicklung der Programmiersprachen ... 43
1.3 Digitale Speicherung und Verarbeitung von Informationen ... 48
1.3.1 Digitale Bilddaten ... 50
1.3.2 Digitale Audiodaten ... 50
1.3.3 Digitale Speicherung von Text ... 51
1.4 Zusammenfassung ... 52

2 Mathematische und technische Grundlagen 53

2.1 Einführung in die Logik ... 53
2.1.1 Aussagen ... 53
2.1.2 Aussageformen ... 54
2.1.3 Logische Verknüpfungen ... 55
2.1.4 Mengenoperationen ... 60
2.2 Informationsspeicherung im Computer ... 64
2.2.1 Bits und Bytes ... 64
2.3 Elektronische Grundlagen ... 73
2.3.1 Einfache Schaltungen ... 73
2.3.2 Zusammengesetzte Schaltungen ... 76
2.4 Automatentheorien und -simulationen ... 79
2.4.1 Algorithmen ... 80
2.4.2 Die Turing-Maschine ... 84

2.4.3 Der virtuelle Prozessor ... 87
2.5 Zusammenfassung ... 93

3 Hardware 95

3.1 Grundlagen ... 95
3.2 Die Zentraleinheit ... 98
3.2.1 Aufbau und Aufgaben des Prozessors ... 99
3.2.2 Der Arbeitsspeicher ... 108
3.2.3 Das BIOS ... 110
3.2.4 Bus- und Anschlusssysteme ... 115
3.3 Die Peripherie ... 123
3.3.1 Massenspeicher ... 124
3.3.2 Eingabegeräte ... 134
3.3.3 Ausgabegeräte ... 136
3.3.4 Sound-Hardware ... 142
3.4 Zusammenfassung ... 143

4 Betriebssysteme 145

4.1 Entwicklung der Betriebssysteme ... 146
4.1.1 Die Geschichte von UNIX ... 147
4.1.2 PC-Betriebssysteme ... 149
4.2 Aufgaben und Konzepte ... 151
4.2.1 Allgemeiner Aufbau von Betriebssystemen ... 151
4.2.2 Prozessverwaltung ... 158
4.2.3 Speicherverwaltung ... 161
4.2.4 Dateisysteme ... 163
4.3 Linux ... 169
4.3.1 Arbeiten mit der Shell ... 171
4.3.2 Wichtige Systembefehle ... 179
4.3.3 Editoren ... 192
4.3.4 Grafische Benutzeroberflächen ... 198
4.3.5 Linux-Installation und -Konfiguration ... 201
4.4 MacOS X ... 206
4.4.1 Mit Aqua arbeiten ... 208
4.4.2 Systemkonfiguration ... 211
4.5 Windows ... 213
4.5.1 Die verschiedenen Windows-Versionen ... 213
4.5.2 Die Windows-Benutzeroberfläche ... 216
4.5.3 Die Windows-Konsole ... 221
4.5.4 Windows-Konfiguration ... 223
4.6 Zusammenfassung ... 226

5 Grundlagen der Programmierung 227

5.1 Die Programmiersprache C 228
5.1.1 Das erste Beispiel 229
5.1.2 Elemente der Sprache C 231
5.1.3 Die C-Standardbibliothek 247
5.2 Java 251
5.2.1 Grundlegende Elemente der Sprache Java 253
5.2.2 Objektorientierte Programmierung mit Java 257
5.2.3 Dateizugriffe in Java 263
5.3 Perl 264
5.3.1 Das erste Beispiel 266
5.3.2 Elemente der Sprache Perl 267
5.4 Zusammenfassung 276

6 Konzepte der Programmierung 277

6.1 Algorithmen und Datenstrukturen 277
6.1.1 Ein einfaches Praxisbeispiel 277
6.1.2 Sortier-Algorithmen 279
6.1.3 Such-Algorithmen 284
6.1.4 Ausgewählte Datenstrukturen 285
6.2 Reguläre Ausdrücke 290
6.2.1 Muster für reguläre Ausdrücke 291
6.2.2 Programmierung mit regulären Ausdrücken 293
6.3 Systemnahe Programmierung 295
6.3.1 Prozesse und Pipes 296
6.3.2 Threads 299
6.4 GUI- und Grafikprogrammierung 303
6.4.1 Zeichnungen und Grafiken erstellen 304
6.4.2 Animation 309
6.4.3 Programmierung fensterbasierter Anwendungen 310
6.4.4 Java-Applets 316
6.5 Die Entwicklungsumgebung Eclipse 317
6.5.1 Überblick 318
6.5.2 Java-Entwicklung mit Eclipse 319
6.6 Zusammenfassung 323

7 Software-Engineering 325

7.1 Überblick 326
7.1.1 Der Entwicklungszyklus 326
7.1.2 Planung und Analyse 327

7.1.3 Entwurf ... 333
7.1.4 Implementierung und Test ... 334
7.1.5 Dokumentation ... 335
7.1.6 Konkrete Entwicklungsverfahren ... 336
7.2 Werkzeuge ... 339
7.2.1 UML ... 339
7.2.2 Entwurfsmuster ... 345
7.2.3 Unit-Tests ... 348
7.3 Zusammenfassung ... 353

8 Datenbanken 355

8.1 Die verschiedenen Datenbanktypen ... 356
8.1.1 Einzeltabellendatenbanken ... 357
8.1.2 Relationale Datenbanken ... 358
8.1.3 Objektorientierte Datenbanken ... 365
8.2 MySQL – ein konkretes DBMS ... 368
8.2.1 MySQL installieren und konfigurieren ... 368
8.2.2 Erste Schritte mit dem mysql-Client ... 372
8.3 SQL-Abfragen ... 372
8.3.1 Datenbanken und Tabellen erzeugen ... 373
8.3.2 Auswahlabfragen ... 377
8.3.3 Einfüge-, Lösch- und Änderungsabfragen ... 380
8.4 Grundlagen der Datenbankprogrammierung ... 382
8.5 Zusammenfassung ... 385

9 Bildbearbeitung und Grafik 387

9.1 Theoretische Grundlagen ... 388
9.1.1 Licht und Farbe ... 388
9.1.2 Bildgröße und Auflösung ... 392
9.2 Bildbearbeitung mit Adobe Photoshop ... 394
9.2.1 Auflösung, Farbmodus und Farbtiefe einstellen ... 395
9.2.2 Auswahlwerkzeuge und -techniken ... 399
9.2.3 Malwerkzeuge und Malfunktionen ... 404
9.2.4 Arbeiten mit Ebenen ... 407
9.3 Bearbeiten von Vektorgrafik mit Adobe Illustrator ... 412
9.3.1 Mit Pfaden arbeiten ... 412
9.3.2 Transformationen und andere Änderungen ... 418
9.4 Zusammenfassung ... 419

10 DTP und digitale Druckvorstufe 421

10.1 Satz, Layout und Typographie ... 421
10.1.1 Grundlagen der Typographie ... 421
10.1.2 Grundlagen der Gestaltung ... 433
10.2 DTP-Grundlagen ... 437
10.2.1 Der DTP-Workflow ... 437
10.2.2 Elektronische Schriften ... 439
10.3 InDesign ... 440
10.3.1 Arbeitsoberfläche und Werkzeuge ... 440
10.3.2 Der InDesign-Arbeitsablauf ... 443
10.4 PDF und Acrobat ... 447
10.4.1 PDF-Dokumente erstellen ... 448
10.5 Zusammenfassung ... 449

11 Multimedia 451

11.1 Einführung ... 451
11.2 Audiobearbeitung mit Adobe Audition ... 453
11.2.1 Einstellungen und Effekte in Audition ... 455
11.2.2 Arrangements erstellen ... 458
11.3 Videoschnitt mit Adobe Premiere ... 459
11.3.1 Die Arbeitsmittel von Premiere ... 460
11.3.2 Einen Film schneiden und erstellen ... 463
11.4 3-D-Grafik und -Animation mit 3ds max ... 467
11.4.1 3ds max im Überblick ... 468
11.4.2 Objekte modellieren ... 470
11.4.3 Materialien zuweisen ... 472
11.4.4 Kameras und Licht hinzufügen ... 472
11.4.5 Einfache Animation ... 473
11.4.6 Rendern ... 474
11.5 Das Autorensystem Macromedia Director ... 474
11.5.1 Director-Grundlagen ... 475
11.5.2 Darsteller erstellen und bearbeiten ... 483
11.5.3 Lingo-Grundlagen ... 494
11.5.4 Director-Filme veröffentlichen ... 511
11.6 Zusammenfassung ... 513

12 Datei- und Datenformate 515

12.1 Textdateien und Zeichensätze ... 515
12.1.1 Das Problem des Zeilenumbruchs ... 516

12.1.2 Zeichensätze ... 518
12.1.3 Textbasierte Dateiformate ... 524
12.2 Binäre Dateiformate ... 527
12.2.1 Bilddateiformate ... 530
12.2.2 Multimedia-Dateiformate ... 533
12.3 Zusammenfassung ... 535

13 Netzwerktechnik 537

13.1 Grundlagen ... 537
13.1.1 Was ist ein Netzwerk? ... 537
13.1.2 Entstehung der Netzwerke ... 539
13.2 Funktionsebenen von Netzwerken ... 543
13.2.1 Das OSI-Referenzmodell ... 543
13.2.2 Das Schichtenmodell der Internetprotokolle ... 545
13.2.3 Netzwerkkommunikation über die Schichten eines Schichtenmodells ... 547
13.3 Klassifizierung von Netzwerken ... 550
13.3.1 Die Reichweite des Netzwerkes ... 551
13.3.2 Die Netzwerktopologie ... 552
13.3.3 Der Zentralisierungsgrad des Netzwerkes ... 553
13.4 Netzwerkkarten, -kabel und Netzzugangsverfahren ... 558
13.4.1 Die verschiedenen Ethernet-Standards ... 558
13.4.2 Token Ring ... 563
13.4.3 Drahtlose Netze ... 564
13.4.4 Sonstige Zugangsarten ... 568
13.5 Datenfernübertragung ... 569
13.5.1 Netzwerkzugang per Modem (analoge Telefonleitung) ... 570
13.5.2 ISDN ... 571
13.5.3 DSL-Dienste ... 573
13.6 Die TCP/IP-Protokollfamilie ... 575
13.6.1 Netzzugang in TCP/IP-Netzwerken ... 576
13.6.2 IP-Adressen, Datagramme und Routing ... 577
13.6.3 Transportprotokolle ... 601
13.6.4 Das Domain Name System (DNS) ... 606
13.6.5 Verschiedene Internet-Anwendungsprotokolle ... 615
13.7 Andere Protokollstapel ... 629
13.7.1 Die AppleTalk-Protokollfamilie ... 629
13.7.2 Novell IPX/SPX ... 632
13.7.3 NetBEUI/SMB ... 633
13.8 Zusammenfassung ... 633

14 Netzwerkanwendungen 635

14.1 Netzwerkkonfiguration unter verschiedenen Betriebssystemen ... 635
14.1.1 Linux ... 635
14.1.2 MacOS X ... 637
14.1.3 Windows ... 639
14.1.4 TCP/IP-Dienstprogramme ... 640
14.2 Server konfigurieren ... 643
14.2.1 Mac OS X ... 643
14.2.2 Windows ... 644
14.2.3 UNIX/Linux ... 647
14.2.4 Der Webserver Apache ... 652
14.3 Einführung in die Netzwerkprogrammierung ... 664
14.3.1 Die Berkeley Socket API ... 664
14.3.2 Ein praktisches Beispiel ... 671
14.4 Verteilte Anwendungen mit J2EE ... 674
14.4.1 Enterprise Java Beans (EJB) ... 675
14.4.2 Java Servlets ... 680
14.4.3 Web Services ... 681
14.5 Zusammenfassung ... 683

15 XML 685

15.1 Der Aufbau von XML-Dokumenten ... 687
15.1.1 Die grundlegenden Bestandteile von XML-Dokumenten ... 687
15.1.2 Wohlgeformtheit ... 693
15.2 DTDs und XML Schema ... 695
15.2.1 Document Type Definitions (DTDs) ... 696
15.2.2 Namensräume ... 705
15.2.3 XML Schema ... 706
15.3 XSLT ... 709
15.4 Grundlagen der XML-Programmierung ... 712
15.4.1 SAX ... 713
15.4.2 DOM ... 716
15.5 Zusammenfassung ... 718

16 HTML, CSS und Webdesign 719

16.1 HTML und XHTML ... 719
16.1.1 Die Grundstruktur von HTML-Dokumenten ... 720
16.1.2 Textstrukturierung und -formatierung ... 722
16.1.3 Listen und Aufzählungen ... 730
16.1.4 Farben in HTML ... 734

16.1.5 Hyperlinks ... 735
16.1.6 Bilder in Webseiten einbetten ... 739
16.1.7 Tabellen ... 742
16.1.8 Formulare ... 749
16.1.9 Einbetten von Multimedia-Dateien ... 753
16.1.10 Frames ... 755
16.1.11 Meta-Tags und Suchmaschinen ... 759
16.2 Cascading Stylesheets (CSS) ... 762
16.2.1 Platzieren von Stylesheets ... 763
16.2.2 Stylesheet-Wertangaben ... 765
16.2.3 Stylesheet-Eigenschaften ... 766
16.2.4 Layer erzeugen und positionieren ... 769
16.3 Grundgedanken zum Webdesign ... 771
16.3.1 Bildschirmauflösung und Farbtiefe ... 772
16.3.2 Die technischen Fähigkeiten der Browser ... 775
16.4 Web-Bildbearbeitung mit Fireworks MX 2004 ... 776
16.4.1 Bilder optimiert für das Web speichern ... 777
16.4.2 Das passende Dateiformat wählen ... 779
16.4.3 Den Export des Bildes vorbereiten und durchführen ... 781
16.4.4 Animierte GIFs erstellen ... 785
16.4.5 Segmente und Verhalten ... 786
16.5 Webdesign mit Macromedia Dreamweaver MX 2004 ... 790
16.5.1 Bearbeiten von HTML-Dokumenten ... 790
16.5.2 Hinzufügen verschiedener Elemente ... 793
16.5.3 Website-Funktionen ... 798
16.5.4 Layers und Verhalten ... 802
16.6 Zusammenfassung ... 804

17 Webanwendungen 807

17.1 Grundlagen ... 807
17.1.1 Konkurrenztechnologien ... 807
17.1.2 Content-Management, Weblogs & Co. ... 808
17.2 PHP ... 810
17.2.1 PHP installieren ... 811
17.2.2 Die PHP-Konfigurationsdatei php.ini ... 814
17.2.3 Die ersten PHP-Beispiele ... 817
17.2.4 PHP-Sprachgrundlagen ... 818
17.2.5 Webspezifische Funktionen ... 830
17.2.6 Gästebücher und Counter mit Textdateien ... 835
17.2.7 Zugriff auf MySQL-Datenbanken ... 845
17.2.8 Zwei größere Beispiele ... 850
17.3 Zusammenfassung ... 852

18 JavaScript 855

18.1 JavaScript-Einführung ... 855
18.2 JavaScript im HTML-Dokument ... 856
18.2.1 Erstes Beispiel: Ausgabe ins Dokument ... 857
18.3 Formulare und Event-Handler ... 861
18.3.1 Zugriff auf Formulare und ihre Elemente ... 862
18.3.2 Kleiner Rechner mit einem Formular ... 863
18.3.3 Die Hintergrundfarbe dynamisch ändern ... 865
18.3.4 Formularauswertung ... 868
18.4 Datums- und Uhrzeit-Funktionen ... 873
18.4.1 Datums- und Uhrzeit-Methoden ... 874
18.4.2 Timeout – die JavaScript-»Stoppuhr« ... 875
18.4.3 Ein ausführliches Beispiel ... 876
18.5 Manipulation von Bildern ... 876
18.5.1 Erstes Beispiel: Austauschen eines Bildes auf Knopfdruck ... 877
18.5.2 Vorausladen von Bildern ... 878
18.5.3 Eine gut funktionierende Rollover-Lösung ... 879
18.5.4 Weitere Beispiele ... 883
18.6 Browser- und Fenster-Optionen ... 888
18.6.1 Browser-Eigenschaften ... 888
18.6.2 Automatische Hyperlinks – History und Location ... 889
18.7 DHTML und DOM ... 896
18.7.1 W3C-DOM ... 897
18.7.2 Das klassische Internet-Explorer-Modell ... 906
18.7.3 Das klassische Netscape-Modell ... 907
18.7.4 Browserübergreifende Lösungen ... 909
18.8 Zusammenfassung ... 914

19 Flash MX 2004 915

19.1 Arbeitsumgebung und Werkzeuge ... 916
19.1.1 Zeichnungen erstellen ... 917
19.1.2 Mit Text arbeiten ... 925
19.1.3 Die restlichen Werkzeuge ... 928
19.2 Animationstechniken ... 928
19.2.1 Elemente und Begriffe der Zeitleiste ... 929
19.2.2 Bewegungs-Tweening ... 932
19.2.3 Maskenebenen verwenden ... 935
19.2.4 Form-Tweening ... 936
19.2.5 Arbeiten mit Symbolen und Instanzen ... 937
19.2.6 Anwendungsbeispiel: Bauanleitung für ein Fahrrad ... 939
19.2.7 Arbeiten mit Farbeffekten ... 942
19.3 Medienintegration ... 942
19.3.1 Sound ... 943

19.3.2 Bitmaps importieren und verwenden ... 944
19.3.3 Digitalvideo ... 945
19.4 Filme exportieren und in Webseiten einbetten ... 946
19.4.1 Überblick über die Exportfunktionen ... 946
19.4.2 Der einfache Export einer SWF-Datei ... 947
19.4.3 SWF-Filme in HTML einbetten ... 948
19.4.4 Die Funktion »Veröffentlichen« ... 951
19.5 Interaktivität mit ActionScript ... 952
19.5.1 Erstes Beispiel: ein Film, der am Ende anhält ... 953
19.5.2 Schaltflächen verwenden ... 954
19.5.3 Navigation ... 955
19.5.4 Andere Filme laden: die loadMovie-Funktionen ... 956
19.5.5 Steuern von Movieclips ... 957
19.5.6 Eigenschaften von Movieclip-Instanzen ... 962
19.5.7 Verschiedene ActionScript-Lösungen ... 963
19.6 Zusammenfassung ... 970

20 Computer- und Netzwerksicherheit 971

20.1 PC-Gefahren ... 972
20.1.1 Viren und Würmer ... 972
20.1.2 Trojaner und Backdoors ... 976
20.1.3 Weitere Schädlinge ... 977
20.2 Netzwerk- und Serversicherheit ... 982
20.2.1 Servergefahren ... 982
20.2.2 Wichtige Gegenmaßnahmen ... 985
20.2.3 Kryptografie ... 986
20.3 Zusammenfassung ... 989

A Glossar 991

B Kommentiertes Literaturverzeichnis 1005

B.1 Allgemeine Einführungen und Überblicke ... 1005
B.2 Mathematische und technische Grundlagen ... 1005
B.3 Hardware ... 1006
B.4 Betriebssysteme ... 1006
B.5 Grundlagen der Programmierung ... 1007
B.6 Konzepte der Programmierung ... 1008
B.7 Software-Engineering ... 1009
B.8 Datenbanken ... 1010
B.9 Bildbearbeitung und Grafik ... 1010

B.10 Desktop Publishing und digitale Druckvorstufe 1011
B.11 Multimedia 1011
B.12 Datei- und Datenformate 1011
B.13 Netzwerktechnik 1012
B.14 Netzwerkanwendungen 1013
B.15 XML 1013
B.16 HTML, CSS und Webdesign 1014
B.17 Webanwendungen 1015
B.18 JavaScript 1015
B.19 Flash MX 2004 1016
B.20 Computer- und Netzwerksicherheit 1017

Index **1019**

Vorwort

Warum beglückt uns die herrliche, das Leben erleichternde, Arbeit ersparende Technik so wenig? Die einfache Antwort lautet: Weil wir noch nicht gelernt haben, einen vernünftigen Gebrauch von ihr zu machen. Im Kriege dient sie dazu, dass wir uns gegenseitig verstümmeln. Im Frieden hat sie unser Leben hastig und unsicher gestaltet. Statt uns weitgehend von geisttötender Arbeit zu befreien, hat sie die Menschen zu Sklaven der Maschine gemacht, die meist mit Unlust ihr eintöniges, langes Tagewerk vollbringen und stets um ihr armseliges Brot zittern müssen.
– Albert Einstein

Kompendium der Informationstechnik – The Next Generation

Eins vorweg: Wie die meisten Geeks[1] bin ich ein Fan der Sciencefiction-Serie »Star Trek«. Sie zeigt nicht nur technisches Spielzeug, das sich jeder Computerfreak wünscht (oder möchten Sie kein Holodeck?), sondern vor allem eine Welt, in der Einsteins Kritik aus dem Motto dieses Kapitels überwunden wurde. Die heutige Welt ist weit von diesem Zustand entfernt, und einige Entwicklungen der letzten Jahre geben Anlass zu der Befürchtung, dass das fortschrittsgläubige »*noch* weit davon entfernt« fehl am Platz sein könnte. Die Menschheit scheint zu dumm oder zu faul zu sein, um drängende, selbst gemachte Probleme in den Griff zu bekommen; Hunger und unmittelbare materielle Not nehmen sogar noch zu.

Dies ist kein Buch über Politik, Wirtschafts- und Sozialwissenschaften, Zukunftsforschung oder Philosophie. Dennoch sollte jeder Mensch – egal, worauf er sich im Leben spezialisiert – die Augen für seine Umgebung und die Welt, in der er lebt, offen halten. Nur interessierte, informierte und gut ausgebildete Menschen werden in der Lage sein, vielleicht doch noch den Weg in die Star Trek-Gesellschaftsordnung zu finden. In der Informations- und Kommunikationstechnik finden sich einige vielversprechende Ansätze dazu. Und zumindest zu einer guten, umfassenden Ausbildung möchte ich mit diesem Buch beitragen.

Wie Sie der Überschrift entnehmen können, ist dieses Buch die zweite Auflage meines Buches »Kompendium der Informationstechnik«. Der neue Titel wurde gewählt, um zu verdeutlichen, dass es noch stärker an die Erfordernisse der IT-Ausbildung angepasst wurde (siehe den Abschnitt »*IT-Ausbildung«* in Kapitel 1, *Einführung*). Die aus der vorigen Auflage übernommenen Inhalte wurden aktualisiert; außerdem sind einige Themen hinzugekommen, darunter die beiden neuen Kapitel über Software-Engineering und IT-Sicherheit.

Wenn Sie die vorige Auflage kennen, werden Sie feststellen, dass dieses Buch trotz der Erweiterung etwas dünner (und daher kostengünstiger) geworden ist. Dies wurde vor

1 Geek, Nerd, Hacker und so weiter: diverse zwischen Respekt und Belustigung oszillierende Bezeichnungen für heillos Computer-Infizierte

allem durch die Auslagerung umfangreicher Listings ins Web und durch einen etwas kompakteren Textsatz erreicht; an einigen Stellen habe ich auch behutsame Kürzungen vorgenommen. Das nunmehr veraltete Thema Mac OS 9 wurde ganz weggelassen – falls Sie dennoch Bedarf haben, finden Sie die entsprechenden Seiten auf der Website zum Buch (**http://buecher.lingoworld.de/fachinfo**).

Und nun wünsche ich Ihnen viel Spaß beim Lesen und Durcharbeiten dieses Buches und viel Erfolg in Ihrer Ausbildung. Ich freue mich auf Ihre Hinweise, Kommentare und Kritik, um dieses Buch noch weiter zu verbessern. Kontaktinformationen erhalten Sie ebenfalls auf der genannten Website.

Worum geht es in diesem Buch?

In diesem Buch werden viele verschiedene Themengebiete behandelt, die mit Computersystemen zu tun haben. Sie lernen zahlreiche Geräte, Betriebssysteme, Programmiersprachen und Anwendungsprogramme kennen und erfahren viel Wissenswertes über Netzwerke und das Internet. Bei allen Themen habe ich versucht, die Balance zwischen Theorie und Praxis zu wahren: Es werden weder die technischen und theoretischen Details verschwiegen, wie in zahlreichen Büchern für absolute Einsteiger, noch kommen die praktischen Anwendungsbeispiele zu kurz, was in manchen akademischen Lehrwerken der Fall ist.

Die einzelnen Kapitel dieses Buches widmen sich den folgenden Themen:

- **Kapitel 1**, *Einführung*, behandelt die Geschichte und grundlegende Funktionsweise des Computers. Außerdem werden einige wichtige Grundlagen der Informationstechnik erläutert; sie bilden die Voraussetzung für das Verständnis späterer Kapitel.
- In **Kapitel 2**, *Mathematische und technische Grundlagen,* werden zunächst die mathematischen und logischen Prinzipien erläutert, auf denen der Computer basiert. Anschließend lernen Sie die wichtigsten elektrotechnischen Grundbausteine kennen. Abgerundet wird das Kapitel durch die Vorstellung von Automatentheorien und Rechnersimulationen.
- **Kapitel 3**, *Hardware*, beschäftigt sich mit den diversen Bauteilen, aus denen ein Computer besteht, sowie mit zahlreichen wichtigen Peripheriegeräten. Sie erfahren nicht nur die wichtigsten technischen Details über Elemente wie den Mikroprozessor, verschiedene Laufwerke und Datenträger oder andere Ein- und Ausgabegeräte, sondern es werden auch praktische Informationen vermittelt. Zum Beispiel werden Einbau und Anschluss von Komponenten oder das BIOS-Setup angesprochen.
- In **Kapitel 4**, *Betriebssysteme*, werden zunächst allgemeine Konzepte erklärt, zum Beispiel die Verwaltung von Prozessen, das Speichermanagement oder die Dateiverwaltung. Anschließend werden die folgenden Betriebssysteme konkret beschrieben: Linux, Mac OS X und Windows.
- In **Kapitel 5**, *Grundlagen der Programmierung*, werden drei verschiedene wichtige Programmiersprachen eingeführt, die unterschiedliche Entwicklungsstufen und Aspekte des Programmierens abdecken: C, Java und Perl.

- **Kapitel 6**, *Konzepte der Programmierung*, baut auf diese Grundlagen auf und erläutert verschiedene Aspekte, die die Programmierung in der Praxis ausmachen: Algorithmen und Datenstrukturen, Reguläre Ausdrücke (mächtige Suchmuster), Elemente der systemnahen Programmierung oder die Entwicklung von Programmen für grafische Oberflächen. Den Abschluss bildet eine Einführung in die freie Entwicklungsumgebung Eclipse.
- Das in **Kapitel 7** behandelte *Software-Engineering* geht weit über reine Programmiertechniken hinaus: Sie lernen viele unterschiedliche Methoden zur Bearbeitung und Verwaltung von Softwareprojekten kennen. Unter anderem werden die Anwendungsmodellierung mit der UML, Grundlagen des Projektmanagements und verschiedene Ansätze des Software-Engineerings wie der Unified Process oder Extreme Programming vorgestellt. Für die in vielen Ausbildungsberufen fällige Projektarbeit sind diese Themen überlebenswichtig.
- **Kapitel 8**, *Datenbanken*, befasst sich mit einer der wichtigsten Funktionsgrundlagen vieler Programmierprojekte und Anwendungen. Nach der üblichen Erläuterung von Begriffen und Konzepten wird als konkretes Datenbanksystem der weit verbreitete Open-Source-Datenbankserver MySQL eingeführt.
- **Kapitel 9**, *Bildbearbeitung und Grafik*, erläutert den Umgang mit Bildern und Zeichnungen am Computer. Für die Bildbearbeitung wird das Programm Photoshop behandelt, als Vektorgrafikprogramm wird Illustrator vorgestellt.
- Thema von **Kapitel 10**, *DTP und digitale Druckvorstufe*, ist die Aufbereitung von Dokumenten für den professionellen Druck. Neben der Satz- und Layout-Praxis mit Adobe InDesign kommen dort auch einige theoretische Grundlagen der Typographie und Gestaltung zur Sprache.
- In **Kapitel 11**, *Multimedia*, geht es um die Herstellung animierter, interaktiver Präsentationen für CD-ROMs, öffentliche Informationsterminals oder Lernsoftware. Zunächst wird die Verarbeitung der wichtigsten »Zutaten« erläutert: Audiobearbeitung mit Adobe Audition, digitaler Videoschnitt mit Adobe Premiere und 3-D-Animation mit 3d s max. Anschließend wird das umfangreiche Autorensystem Macromedia Director vorgestellt, mit dem diese Bestandteile zur fertigen Präsentation zusammengebaut werden.
- **Kapitel 12**, *Datei- und Datenformate*, widmet sich den wichtigsten Formaten für Text, Bild und Multimedia. Es geht zunächst um Text und Zeichensätze, anschließend werden verschiedene textbasierte und binäre Dateiformate konkret erläutert.
- In **Kapitel 13**, *Netzwerktechnik*, wird die Entwicklung der Netzwerke und des Internets beschrieben, und Sie erhalten eine Einführung in die Begriffswelt der Netzwerke. Beispielsweise werden Schichtenmodelle und Netzwerkarchitekturen vorgestellt. Danach wird die genaue Funktionsweise verschiedener Arten von Netzwerkkarten und -anschlüssen erläutert; außerdem werden diverse Netzwerkprotokolle beschrieben. Den Schwerpunkt bildet die TCP/IP-Protokollfamilie, die für das Internet entwickelt wurde und inzwischen der wichtigste Kommunikationsstandard für alle Arten von Netzwerken ist.
- In **Kapitel 14**, *Netzwerkanwendungen*, erhalten Sie einen Überblick über einige wichtige Aspekte der praktischen Verwendung von Netzwerken: Sie erfahren, wie Sie die Netzwerkunterstützung in den drei weiter oben genannten Betriebssystemen aktivieren und

konfigurieren, wie Sie einige wichtige Serverdienste in Betrieb nehmen können und wie Netzwerkanwendungen programmiert werden. Den Abschluss bildet eine Einführung in die Programmierung verteilter Anwendungen mit der Java 2 Enterprise Edition und in Web Services.

- **Kapitel 15**, *XML*, stellt die eXtensible Markup Language vor, eine Sprache, die der Definition beliebiger hierarchisch gegliederter Dokumentformate dient. In zahlreichen Anwendungen wird XML inzwischen eingesetzt, so dass es nützlich ist, die Konzepte dieses Formats zu kennen. Sie erfahren das Wichtigste über wohlgeformte Dokumente, über Formatbeschreibungen mit Hilfe von DTDs und XML Schema, die Umwandlung von XML-Dokumenten mit XSLT sowie über die Programmierung XML-basierter Anwendungen.
- In **Kapitel 16**, *HTML, CSS und Webdesign*, wird HTML beziehungsweise XHTML vorgestellt, die Sprache, in der Webseiten verfasst werden. Hier werden verschiedene konkrete Aspekte der Webseitenerstellung erläutert, etwa die Text- und Schriftformatierung, der Listen- und Tabellensatz, das Einbetten von Bildern, Hyperlinks und Webformulare. Anschließend werden einige wichtige Grundregeln für die Gestaltung von Webseiten vermittelt, und es werden zwei konkrete Programme vorgestellt, mit denen Sie diese Arbeit erledigen können: das Web-Bildbearbeitungsprogramm Fireworks und der grafisch orientierte Webseiten-Editor Dreamweaver.
- In **Kapitel 17**, *Webanwendungen*, erfahren Sie, wie man Websites erstellt, die nicht nur aus statischen HTML-Dokumenten, sondern auch aus dynamisch generierten Inhalten bestehen. Konkret wird die Programmiersprache PHP 5 behandelt. Neben den allgemeinen Konzepten wird die Programmierung konkreter Anwendungen beschrieben: Sie erfahren, wie man Gästebücher, Counter oder Diskussionsforen schreibt, und zwar sowohl auf der Basis von Textdateien als auch mit Hilfe einer (MySQL-)Datenbank.
- **Kapitel 18**, *JavaScript*, stellt die wichtigste clientseitige Programmiersprache vor, mit der Sie die Inhalte einer Webseite »zum Leben erwecken« können. Zunächst werden die klassischen Anwendungen wie die Ausgabe ins Dokument selbst, die Verarbeitung von Formularen oder der Austausch von Bildern behandelt, anschließend erfahren Sie das Wichtigste über »Dynamic HTML« – eine Sammlung von Techniken, mit deren Hilfe Sie die Elemente eines Dokuments nachträglich modifizieren können.
- In **Kapitel 19**, *Flash MX 2004*, wird das gleichnamige Programm vorgestellt, mit dessen Hilfe Sie Animationen und interaktive Anwendungen für den Webeinsatz auf der Basis von Vektorgrafik erstellen können. Nach einer ausführlichen Einführung in Zeichen- und Animationstechniken lernen Sie die wichtigsten Aspekte der eingebauten Programmiersprache ActionScript kennen.
- **Kapitel 20**, *Computer- und Netzwerksicherheit*, befasst sich mit verschiedenen Themen der immer wichtiger werdenden Sicherheit: Der Schutz vor Viren, Würmern und Trojanern wird ebenso behandelt wie Kryptographie, Firewalls oder Fragen der Datensicherheit.

Wie Sie bereits an dieser Aufzählung bemerken, besitzt dieses Buch einen Schwerpunkt: Die Netzwerk- und Internetthemen werden ausführlicher behandelt als andere Aspekte der Informationstechnik. Das liegt nicht nur daran, dass sie seit Jahren den Schwerpunkt meiner Arbeit als Trainer bilden, sondern auch daran, dass sie besonders wichtig sind: Das

Internet hat die Art und Weise, wie mit einem Personal Computer gearbeitet wird, erheblich stärker revolutioniert, als Ihnen bewusst sein könnte. Außerdem werden diese Techniken im Wesentlichen auch dann noch weiter existieren, wenn der PC als dominierendes Gerät für den Internetzugang durch kleinere, leichter zu bedienende Mobilgeräte ersetzt werden sollte, wie das einige vermuten.

Wichtiger Hinweis

Viele Anwendungsprogramme, die hier vorgestellt werden, sind in Versionen für Windows und Mac OS erhältlich. Da ich das Buch an einem Windows-PC geschrieben habe, wurden auch alle beschriebenen Arbeitsschritte an diesem Rechner ausprobiert und nachvollzogen; auch die Screenshots der Programme stammen vom PC. Aus diesem Grund hier ein wichtiger Hinweis für die Mac-Anwender: Fast alle Tastenkürzel, die unter Windows mit der Taste Strg gebildet werden, erzeugt man auf dem Mac stattdessen mit der ⌘-Taste (Befehl- oder Apfel-Taste). Soweit dies der Fall ist, habe ich entsprechende Hinweise weggelassen. Der Text eines Fachbuches wird nämlich gewiss nicht besser lesbar, wenn Sie sich andauernd durch Konstrukte wie »Drücken Sie Strg + B beziehungsweise ⌘ + B ...« quälen müssen.

Was Sie hier nicht finden

Trotz des immensen Seitenumfangs kann in einem Buch wie diesem nicht jedes Thema behandelt werden. Ich hoffe, Sie finden die Auswahl genauso sinnvoll wie ich. Zu den wichtigsten Inhalten, die nicht in diesem Buch behandelt werden, gehören folgende:

- **Nichttechnische Ausbildungsinhalte**. Die Ausbildung zum Fachinformatiker, Mediengestalter oder für andere IT-Berufe enthält neben den fachspezifischen Inhalten auch Aspekte wie Wirtschafts- und Sozialwissenschaften, Rechtskunde oder Betriebsorganisation. Solche Themen können in diesem Buch nicht berücksichtigt werden.
- **Office-Anwendungen**. Anleitungen zu Word, Excel & Co. gibt es wirklich genügend; der Platz in diesem Buch reicht nicht aus, um auch noch darauf einzugehen. Nichtsdestotrotz ist gerade die Tabellenkalkulation Excel ein sehr praktisches und spannendes Programm – zum Beispiel, um die mathematischen Grundlagen aus Kapitel 2 nachvollziehen zu können.
- **Speziellere Anwendungsprogramme**. Wie Sie an der Kapitelübersicht oben gesehen haben, werden in diesem Buch zahlreiche Anwendungen besprochen. Allerdings gibt es viele Branchen- und Spezialistenlösungen, die in einem so allgemeinen Buch wie diesem nichts zu suchen haben. Beispiele sind etwa CAD-Software, MIDI-Sequencer (mit denen man nur dann arbeiten kann, wenn man eben Musik machen kann) oder Warenwirtschaftssysteme. Gerade zu Letzteren, genauer gesagt zu SAP R/3, existiert ohnehin ein kompletter Verlagsbereich bei Galileo Press.

Für wen ist dieses Buch geeignet?

In erster Linie können Sie mit diesem Handbuch etwas anfangen, wenn Sie eine Ausbildung im IT-Bereich oder verwandten Berufen absolvieren. Es wurde insbesondere für den im Titel genannten Ausbildungsberuf des Fachinformatikers in den beiden Fachrichtungen Anwendungsentwicklung und Systemintegration geschrieben, dürfte aber auch für IT-Systemelektroniker oder IT-Systemkaufleute gut geeignet sein. Genau wie in Ihrer Ausbildung werden auch in diesem Buch viele verschiedene Themenbereiche behandelt: Da Computer sehr komplexe Maschinen sind, ist es erforderlich, sich selbst dann ein großes Spektrum verschiedener Kenntnisse anzueignen, wenn Sie nur in einem bestimmten Fachbereich arbeiten möchten.

Eine weitere Gruppe von Ausbildungsberufen, für die die Arbeit mit diesem Buch sinnvoll sein dürfte, sind die diversen Ausprägungen des Berufsbildes Mediengestalter für Digital- und Printmedien. Selbstverständlich erfolgt die Behandlung der Grafik-, Design- und DTP-Themen hier nicht in der Ausführlichkeit, die Sie benötigen. Aber zu Ihrer Ausbildung gehören eben auch Themen wie allgemeine EDV, Netzwerktechnik oder grundlegende Informationen über Datenbanken. Darüber hinaus macht sich gerade auf einem (leider) engen Arbeitsmarkt jedes zusätzliche Wissen bezahlt: Gerade kleinere Betriebe können sich keine Vollzeit-Administratoren leisten und stellen bevorzugt Mitarbeiter ein, die neben ihrer gestalterischen Arbeit auch die Computertechnik selbst beherrschen.

Auch für Studenten im Grundstudium der Informatik oder in den Informatik-Kursen anderer Studienrichtungen ist das Buch durchaus geeignet. Gerade der Mittelweg zwischen Darstellung der theoretischen Grundlagen und praktischer Anleitung dürfte für Sie eine wichtige Lücke schließen.

Zu guter Letzt ist dieses Buch aber natürlich auch für alle anderen geeignet, die an Computern, Programmierung oder Netzwerken interessiert sind. Es eignet sich nicht nur als Unterrichtsbegleiter, sondern auch zum Selbststudium einzelner Themen. Die einzige Voraussetzung ist im Grunde genommen, dass Sie einen Computer zur Verfügung haben und grundsätzlich wissen, wie man damit umgeht.

Zum Komplexitätsniveau ist noch anzumerken, dass dies weder ein Buch für absolute Neueinsteiger noch ein Begleiter für vollkommene Experten ist. Wenn Sie noch nie mit einem Computer gearbeitet haben, benötigen Sie eine grundlegendere Anleitung; im vorliegenden Buch erfahren Sie nicht, wie man das Gerät in Betrieb nimmt, seine Arbeit als Datei speichert, einen Ordner anlegt oder einen Webbrowser bedient. All diese Dinge (und noch einige mehr) müssen klar sein, bevor Sie etwas Sinnvolles mit diesem Handbuch anfangen können. Wenn Sie dagegen bereits Erfahrung haben, gibt es wahrscheinlich einige Themen, über die Sie noch nicht Bescheid wissen. In diesem Fall finden Sie möglicherweise hier, was Sie suchen.

Icons

Um das Buch für Sie übersichtlicher und damit noch nützlicher zu machen, finden Sie an manchen Stellen die folgenden drei Symbole in der Randspalte:

Dieses Icon markiert einen hilfreichen Tipp, eine Abkürzung oder eine nützliche Zusatzinformation.

Mit diesem Symbol werden Warnungen gekennzeichnet. Sie sollten sie ernst nehmen, um keine unnötigen Datenverluste oder Fehlfunktionen zu erleiden.

Um dieses Buch nicht noch dicker zu machen, habe ich einige besonders umfangreiche Programmlistings ins Web ausgelagert. Auf der Website zum Buch (**http://buecher.lingoworld.de/fachinfo**) brauchen Sie nur den Link **Listings** anzuklicken, und Sie finden die entsprechend nummerierten Listings sowohl zum Online-Lesen als auch gezippt zum Download.

IHK-Fragen und Beispiellistings im Internet

Unter **http://www.buecher.lingoworld.de/fachinfo** finden Sie nicht nur die genannten langen Listings, sondern auch die im Buch abgedruckten Beispiele, so dass Sie diese nicht abtippen müssen. Außerdem finden Sie dort Fragen, mit denen Sie sich auf die IHK-Prüfung vorbereiten können, sowie jeweils aktualisierte Literatur- und Linklisten.

Danksagungen

Zunächst einmal, wie immer, vielen Dank an meinen Lektor Stephan Matteschеck, an die Korrektoren und an das Team von Galileo Computing. Dies ist bereits das dritte gemeinsame Projekt, und aufgrund der stetigen Ermutigung und unermüdlichen Geduld des Verlags macht die Zusammenarbeit von Buch zu Buch noch mehr Spaß.

Vielen Dank auch an die technischen Gutachter der beiden neuen Kapitel, Ralf Bokelberg sowie Steffen Wendzel und Johannes Plötner, für ihre wertvollen Hinweise.

Ich danke auch den zahlreichen Lesern der ersten Auflage, die mir nette Kommentare und auch Hinweise auf Fehler zugeschickt haben. Durch diesen wertvollen Beitrag konnte ich viele Stellen noch entscheidend verbessern oder klarer formulieren. Die fast durchweg positive Resonanz auf die Vorauflage – das »Kompendium der Informationstechnik« – hat nicht nur zu meinem Wohlbefinden beigetragen, sondern auch die Motivation zum Schreiben des vorliegenden Buches gesteigert ;-).

Eine Danksagung wäre nicht komplett, wenn sie nicht auch die wichtigsten Menschen in meinem Leben einschließen würde: meine Familie. Vor allem mein Sohn und meine Frau haben wieder einmal unzählige Stunden auf mich verzichten müssen und waren selbst immer für mich da.

1 Einführung

In der Informatik geht es genauso wenig um Computer, wie in der Astronomie um Teleskope.
– Edsger W. Dijkstra

In diesem Kapitel erhalten Sie einen Überblick über die wichtigsten Grundlagen der Informationstechnik: eine Abgrenzung der verschiedenen Fachrichtungen und Ausbildungsgänge der Informatik, die Entwicklungsgeschichte der Computer sowie eine allgemeine Übersicht über die digitale Speicherung verschiedener Arten von Informationen.

1.1 Informationstechnik, Informatik und EDV

Allgemein gesprochen geht es in diesem Buch um **Informationstechnik** (englisch: Information Technology oder kurz IT). Der traditionelle Begriff für diese Art der Technik lautet **Elektronische Datenverarbeitung** (EDV). **Daten** oder **Informationen** sind Werte, die im Zusammenhang mit beliebigen Sachverhalten angelegt werden oder die im Rahmen von Mess- oder Rechenvorgängen anfallen. **Datenverarbeitung** ist der Vorgang der Sammlung, Speicherung und Manipulation dieser Informationen. Im Lauf der Zeit haben sich unterschiedliche Verfahren der Datenverarbeitung entwickelt:

Arten der Datenverarbeitung

- Die **manuelle Datenverarbeitung** führt Berechnungen und Datenmanipulationen ohne jegliche Hilfsmittel durch; sie basiert auf Kopfrechnen und Auswendiglernen. Das äußerste erlaubte Hilfsmittel ist ein Schreibblock, um Daten oder Zwischenergebnisse zu notieren.
- Die **mechanische Datenverarbeitung** verwendet mechanische Hilfsmittel für die Verarbeitung von Informationen, beispielsweise eine mechanische Schreibmaschine oder einen Rechenschieber.
- Bei der **elektrischen Datenverarbeitung** werden elektrisch betriebene Geräte als Hilfsmittel eingesetzt, zum Beispiel elektrische Schreibmaschinen oder Registrierkassen.
- Die **elektronische Datenverarbeitung** verwendet schließlich elektronisch gesteuerte Arbeitsmittel, also Elektronenrechner oder Computer.

1.1.1 Fachrichtungen der Informatik

Die wissenschaftliche Fachrichtung, die sich mit den verschiedenen Aspekten der Computertechnik auseinander setzt, wird seit den 60er-Jahren des 20. Jahrhunderts als **Informatik** (englisch: Computer Science) bezeichnet. Die akademische Informatik wird üblicherweise in vier Fachrichtungen unterteilt:

- Die **theoretische Informatik** betrachtet insbesondere die mathematisch-logischen Grundlagen, die der Verwendung und Programmierung von Computern zugrunde liegen. Es geht beispielsweise um die **Berechenbarkeit** (ist ein Problem überhaupt durch Berechnung lösbar?) und um Automatentheorien – die mathematisch-formalen Modelle, auf denen Rechner unabhängig von der elektronischen Machbarkeit aufbauen.

- Die **technische Informatik** beschreibt die elektronisch-technischen Eigenschaften der Bauteile, aus denen Computer zusammengesetzt sind. Ein wichtiges Teilgebiet der technischen Informatik ist die **Schaltalgebra**, die Umsetzung logischer Operationen durch elektronische Schaltungen.
- In der **praktischen Informatik** geht es im Großen und Ganzen um die **Programmierung** von Computern und die Mittel, die dazu erforderlich sind. Die Erforschung des Aufbaus von Betriebssystemen und Programmiersprachen-Compilern sowie deren Implementierung (praktische Umsetzung) sind die wichtigsten Teilgebiete.
- Die **angewandte Informatik** kümmert sich gewissermaßen um alles andere, nämlich um sämtliche Nutzanwendungen von Computern. Das reicht von Datenbanken über die Netzwerkkommunikation bis hin zu Grafik, Animation und Audio-/Videobearbeitung.

Da es sich bei diesem Buch um ein Praxisbuch handelt, das nicht für das trockene Auswendiglernen von Lehrsätzen geschrieben wurde, sondern für die alltägliche Nutzung von Computern, ist es kaum verwunderlich, dass sich fast alle Kapitel mit Aspekten der angewandten Informatik beschäftigen. In den Kapiteln 4 bis 7 werden auch die wichtigsten Ansätze der praktischen Informatik, nämlich Betriebssysteme und diverse Themen der Programmierung, behandelt.

Einige grundlegende Aspekte der theoretischen Informatik lernen Sie in Kapitel 2, *Mathematische und technische Grundlagen*, kennen: Dort werden die wichtigsten mathematischen und logischen Operationen besprochen, die Computer ausführen. Außerdem wird beispielhaft auf die Realisierung einiger dieser Funktionen durch elektronische Bauteile eingegangen, also auf einige Ansätze der technischen Informatik.

Im Übrigen gibt es noch ein eigenes Kapitel zum Thema Hardware, die ebenfalls dem Gebiet der technischen Informatik zugeordnet werden kann (Kapitel 3).

1.1.2 Überblick über die IT-Ausbildung

Das Berufsfeld der Informationstechnik und der Mediengestaltung ist vielfältig. Deshalb gibt es zahlreiche unterschiedliche Möglichkeiten einer Ausbildung in diesem Bereich. Sie lassen sich zum einen nach Fachgebieten wie Anwendungsentwicklung, kaufmännischem EDV-Einsatz oder Mediengestaltung unterscheiden. Zum anderen gibt es zwei grundlegende Ausbildungsformen: Berufsausbildung und Studium.

Überblick Ausbildungsgänge

Aus diesen beiden Unterteilungen ergeben sich unter anderem folgende konkreten Ausbildungsgänge:

- Ausbildungsberufe
  - Fachinformatiker
  - IT-Systemelektroniker
  - IT-Systemkaufmann
  - Informatikkaufmann
  - Mediengestalter für Digital- und Printmedien
  - Mediengestalter Bild/Ton

- Studiengänge
  - Informatik
  - Informatik (FH)
  - Wirtschaftsinformatik
  - Medieninformatik
  - Bioinformatik
  - Medizinische Informatik

Ausbildungsberufe

Bis vor wenigen Jahren galt die Informationstechnik als zu komplex, um in einer praxisorientierten Berufsausbildung im klassischen dualen System (Ausbildungsbetrieb – Berufsschule) gelehrt zu werden. Die beiden angebotenen Ausbildungen zum EDV-Kaufmann und Büromaschinenelektroniker hatten andere Schwerpunkte, nämlich einen kaufmännischen beziehungsweise elektrotechnischen.

IT-Berufe

Erst 1996 wurden die IT-Berufe in der Bundesrepublik durch ein Übereinkommen von Arbeitgeberverbänden und Gewerkschaften unter Beratung des Bundesinstituts für Berufsbildung (BiBB) neu geordnet. Seitdem gibt es vier verschiedene IT-Ausbildungsberufe. Sie bereiten auf Tätigkeiten im Bereich der Informations- und Telekommunikationstechnik vor, die früher Bewerbern mit Studienabschluss vorbehalten waren oder in Einzelfällen an qualifizierte Quereinsteiger vergeben wurden.

Im Einzelnen handelt es sich um folgende Ausbildungsgänge:

- Fachinformatiker (Fachrichtungen Anwendungsentwicklung und Systemintegration)
- IT-Systemelektroniker
- IT-Systemkaufmann
- Informatikkaufmann

Fachinformatiker

Aufgabe der **Fachinformatiker** ist es gemäß offizieller Definition, »fachspezifische Anforderungen in komplexe Hard- und Softwaresysteme« umzusetzen. Diese recht ungenaue Beschreibung läuft in der Praxis auf sehr vielfältige Anforderungen hinaus, da sich nicht nur die beiden Fachrichtungen Anwendungsentwicklung und Systemintegration, sondern vor allem auch Ausbildungsbetriebe und Einsatzumgebungen oft stark voneinander unterscheiden. Gemeinsam ist ihnen allen lediglich der Umgang mit Computersystemen, mit Netzwerken, mit der unterschiedlichsten Software und dem komplexen Zusammenspiel dieser Komponenten.

Die beiden Berufsbilder sollen sich im Laufe der drei Ausbildungsjahre allmählich auseinander entwickeln. Bei Ausbildungsbeginn stehen gemeinsame Grundlagen im Vordergrund. Dazu gehören nicht nur fachspezifische Themen, wie sie in diesem Buch behandelt werden, sondern auch wichtige Informationen zum Arbeitsablauf und zum betrieblichen Umfeld. Dies sind insbesondere Grundlagen der Betriebswirtschaftslehre, der Projekt- und Betriebsorganisation, des Arbeits- und Ausbildungsrechts sowie der betrieblichen Buchführung. Diese Kenntnisse werden vor allem in der Berufsschule (60 Tage pro Ausbildungsjahr) vermittelt.

Anwendungsentwicklung

Der Schwerpunkt des **Fachbereichs Anwendungsentwicklung** ist die Erstellung von Software für den eigenen Betrieb oder für Kunden. Der erste Schritt ist die Entwicklung neuer oder die Anpassung vorhandener Programme gemäß den Anforderungen der späteren Anwender. Es folgt die Einrichtung der Software auf den gewünschten Systemen, die gegebenenfalls dafür angepasst werden müssen. Anschließend sollen die Benutzer informiert oder gar geschult werden. In das Umfeld des Berufsbildes gehört demzufolge auch die Erstellung brauchbarer Dokumentationen für Entwickler, Administratoren und Endanwender.

Systemintegration

Im **Fachbereich Systemintegration** liegen die Hauptaufgaben im Bereich der Hardware-, Software- und Netzwerkeinrichtung. Hier sind gute Kenntnisse der verschiedenen Hardwarekomponenten, Betriebssysteme sowie der Netzwerkgeräte und -dienste gefragt. Auch die Schnittstellen zwischen der Informations- und der Telekommunikationstechnik spielen in diesem Beruf eine wichtige Rolle.

IT-Systemelektroniker

Der Beruf des **IT-Systemelektronikers** ist elektrotechnischer orientiert als der des Fachinformatikers. Mehr als beim Fachinformatiker der Fachrichtung Systemintegration liegt der Schwerpunkt hier im Bereich der Hardware. Zu den wichtigsten Aufgaben gehört die Verkabelung von Computern und Netzwerken, angefangen bei der Stromversorgung über den Anschluss von Peripheriegeräten bis hin zu Besonderheiten der Netzwerkinfrastruktur. Dazu benötigen IT-Systemelektroniker ein umfangreiches Know-how über Hardwarekomponenten und Treiber sowie über deren Zusammenarbeit mit Betriebssystemen und Anwendungsprogrammen.

IT-Systemkaufleute

IT-Systemkaufleute sind im Wesentlichen die Kundenberater für Hard- und Softwareprojekte. Sie verfügen über technisches Know-how, so dass sie die Einzelheiten von IT-Projekten verstehen und den Kunden erklären können, und benötigen umfangreiche betriebswirtschaftliche und kaufmännische Kenntnisse, um die Kalkulation für ein solches Projekt durchführen zu können. Sie sind für alle Phasen der Kundenbetreuung für IT-Lösungen zuständig: angefangen bei der Marktanalyse über Beratung und Marketing bis hin zum Vertrieb sowie Einarbeitung und Schulung.

Informatikkaufleute

Bei den **Informatikkaufleuten** stehen die kaufmännischen Aufgaben stärker im Vordergrund, während die technischen Aspekte etwas weniger ausgeprägt sind. Sie bilden das Bindeglied zwischen wirtschaftlichen und technischen Abteilungen. Typischerweise vermitteln sie zwischen den Kunden, die vor allem wirtschaftliche Fragen haben, und den Beratern der Hard- und Softwarehersteller (oft den oben genannten IT-Systemkaufleuten), die eine eher technische Sicht auf Vorgänge benötigen. Beispielsweise könnten Informatikkaufleute die technische Umsetzung von Geschäftsprozessen anleiten und entsprechende Pflichtenhefte erstellen. Eine weitere Aufgabe wäre es, die Unterschiede verschiedener IT-Lösungen in kaufmännisch kalkulierbaren Zahlen auszudrücken.

Alle IT-Berufe beinhalten eine (ausschließlich theoretische) Zwischenprüfung. Es besteht Teilnahmepflicht, aber das Ergebnis ist ausschließlich informativ – sowohl für die Auszubildenden als auch für die Betriebe. Die Abschlussprüfung besteht aus einer selbst gewählten Projektarbeit, die von der jeweils zuständigen Industrie- und Handelskammer genehmigt werden muss, einer Präsentation dieser Arbeit mit zusätzlichen offenen, mündlichen Fragen der Prüfungskomission sowie einem schriftlichen Teil mit allgemeinen Fragen zur Theorie.

Mediengestalter

Neben den genannten IT-Berufen ist dieses Buch auch für Auszubildende der Medienbranche nützlich. Auch diese Ausbildungsgänge wurden vor einigen Jahren neu geordnet und in den verschiedenen Fachrichtungen der **Mediengestalter für Digital- und Printmedien** zusammengefasst. Dabei wurden altehrwürdige Berufsbilder wie Schriftsetzer, Lithograph oder Druckvorlagenhersteller mit neuen Anforderungen der digitalen Medientechnik kombiniert. Das ist nicht nur vorteilhaft – mitunter macht es einen ähnlichen Eindruck, als würde man Berufe wie Architekt, Statiker, Maurer, Schreiner, Elektriker und Installateur zum »Gebäudehersteller« zusammenfassen.

Die Berufsausbildung kann in vier verschiedenen Fachrichtungen mit gemeinsamen Grundlagen und leicht unterschiedlicher Spezialisierung (vor allem im letzten der drei Ausbildungsjahre) absolviert werden:

- In der Fachrichtung **Medienberatung** liegt das Hauptaugenmerk auf der Zusammenarbeit mit den Kunden: Ihre Anforderungen müssen durch die fachliche Unterstützung der Medienberater ein konkretes Profil ergeben, nach welchem deren gestalterisch und technisch orientierte Kollegen arbeiten können. Dieses Profil muss zudem für beide Seiten annehmbar kalkuliert werden, so dass Medienberater auch die umfangreichsten wirtschaftlichen Kenntnisse der Branche benötigen.
- Die Fachrichtung **Mediendesign** kümmert sich um die konkrete Umsetzung der Kundenanforderungen und des daraus entwickelten Profils. Sie erarbeiten Entwürfe – zunächst meist zeichnerisch, ohne Rechnereinsatz – und setzen diese dann mit Hilfe geeigneter Spezialsoftware um. Sie benötigen vor allem ein gutes Auge für Formen und Farben, hervorragende Kenntnisse der Typografie und Gestaltung sowie ein sicheres Gespür für die Aussagewirkung der gewählten Gestaltungsmittel.
- Die Hauptaufgabe der Fachrichtung **Medienoperating** ist die Erstellung ausgabefertiger Endprodukte aus den Entwürfen und Dateien der Mediendesigner. Medienoperatoren müssen insbesondere über Datei- und Datenformate, Farbkorrektur und Druckverfahren Bescheid wissen.
- Mediengestalter der Fachrichtung **Medientechnik** beschäftigen sich vor allem mit den speziellen Geräten und Techniken der Medienherstellung: Sie kennen die verschiedensten Scan-, Druck- und Reproduktionsverfahren, so dass sie den Mediendesignern digitalisierte Vorlagen bereitstellen und die fertigen Dateien der Medienoperatoren für die Endausgabe vorbereiten.

Auszubildende können in jeder der vier Fachrichtungen zwischen den beiden Schwerpunkten Digital- oder Printmedien wählen. Der Begriff Digitalmedien umfasst alle Arten von Multimedia-Präsentationsformen wie digitale Diashows, CD- und DVD-ROM sowie – vor allem – das Web. Printmedien sind sämtliche durch Druckverfahren hergestellten Informationsträger wie Broschüren, Briefbögen, Visitenkarten, Plakate oder Zeitschriften.

Etwas abseits von den anderen Mediengestalter-Ausbildungen steht der Beruf des **Mediengestalters Bild/Ton**. Hier geht es speziell um Verfahren der Audio- und Videobearbeitung, wie sie für Multimedia-Produktionen, Werbespots oder auch allgemein im Film- und Fernsehbereich benötigt werden.

Auch bei den Mediengestaltern gibt es nach der halben Ausbildungsdauer eine rein informative Zwischenprüfung. Die Abschlussprüfung besteht aus zwei Teilen: Zunächst muss eine Praxisarbeit erstellt werden; für die Projektplanung gibt es etwa zwei Wochen Zeit, während die eigentliche Durchführung an ein bis zwei Tagen im Ausbildungsbetrieb stattfinden soll. Für die vier Fachrichtungen und die beiden Schwerpunkte sind dabei unterschiedliche Aspekte desselben Gesamtprojekts vorgesehen. Der zweite Teil ist eine schriftliche Prüfung, in der die verschiedensten offenen Fragen ausführlich beantwortet werden müssen.

Ausbildung und dieses Buch

In Tabelle 1.1 sehen Sie, welche Kapitel dieses Buches für die Auszubildenden der einzelnen Fachrichtungen besonders nützlich (x) beziehungsweise unerlässlich (+) sind. Wenn ein Kapitel für Ihren Beruf nicht angekreuzt wurde, bedeutet dies nicht, dass Sie sich gar nicht mit diesem Thema beschäftigen sollten! Wie bereits im Vorwort erwähnt, ist ein möglichst gründliches Allgemeinwissen eine der wichtigsten Voraussetzungen für beruflichen Erfolg. Zudem verstehen Sie Zusammenhänge der IT- und Medienwirtschaft umso besser, je größer Ihr Überblick ist.

Kapitel	Fachinf. Anwpend.	Fachinf. Systemint.	Kaufl.[1]	IT-Systemelektr.	MG Digital	MG Print	MG Bild/ Ton
1	+	+	+	+	+	+	+
2	+	+	x	+	x	x	x
3	x	+	x	+	x	x	x
4	+	+	x	+	x		
5	+	+	x	+	x		
6	+	x		x			
7	+	+	+	+			
8	+	+	x	x			
9	x				+	+	+
10					+	x	x
11	x				+	x	+
12	+	+	x	x	+	x/+[2]	x
13	+	+	x	+	x	x	x
14	x	+		x	x		

Tabelle 1.1 Die einzelnen Kapitel dieses Buches und ihre Relevanz für verschiedene Ausbildungsberufe

1 Betrifft im Wesentlichen sowohl IT-Systemkaufleute als auch Informatikkaufleute.
2 Schwerpunktthema der Fachrichtung Medienoperating

Kapitel	Fachinf. Anwpend.	Fachinf. Systemint.	Kaufl.[1]	IT-Systemelektr.	MG Digital	MG Print	MG Bild/ Ton
15	+	+			x	x	
16	+	x	x	x	+	x	x
17	+	x			x		
18	+	x			x		
19	+	x			+	x	x
20	x	+	x	+	x	x	x

Tabelle 1.1 Die einzelnen Kapitel dieses Buches und ihre Relevanz für verschiedene Ausbildungsberufe (Forts.)

Studiengänge

Die Studiengänge im Bereich der Informationstechnik sind wesentlich gründlicher, dafür aber auch viel theoretischer orientiert als die betrieblichen Ausbildungsgänge. Neben den hier behandelten Studienfächern der Informatik gibt es übrigens auch angrenzende Fachgebiete wie Telematik, Elektrotechnik oder angewandte Mathematik.

Informatik

Das allgemeinste Spektrum bietet das Studium der **Informatik**. Das Grundstudium beinhaltet Einführungen in alle vier Disziplinen des Faches (theoretische, technische, praktische und angewandte Informatik). Im Hauptstudium ist dagegen eine Spezialisierung auf einen bestimmten Bereich innerhalb eines dieser Zweige vorgesehen. Denkbar wären etwa Betriebssysteme aus dem Gebiet der praktischen Informatik oder Datenbanken, die zur angewandten Informatik zählen.

Das Informatikstudium wird sowohl an Universitäten als auch an Fachhochschulen angeboten. Naturgemäß ist Letzteres etwas praktischer orientiert; das Lernen erfolgt besonders im Grundstudium schulähnlicher. In beiden Lehreinrichtungen ist der typische Abschluss das klassische Diplom; dazu muss eine Diplomarbeit über ein bestimmtes, selbst gewähltes Forschungsthema verfasst werden. Daneben bieten immer mehr Hochschulen und Fachhochschulen auch internationale Abschlüsse wie Bachelor und Master an, deren Anerkennung durch (insbesondere deutsche) Arbeitgeber allerdings noch im Zweifel steht.

Nebenfächer

Wie in jedem naturwissenschaftlichen Studium müssen auch bei der Informatik Nebenfächer belegt werden. Die Auswahl ist so vielfältig wie die Anwendungsbereiche der Informationstechnik: Für theoretisch ausgerichtete Forschungen wäre etwa die Mathematik interessant; ein Fach aus dem Bereich der Wirtschaftswissenschaften eröffnet den Zugang zur Entwicklung kaufmännischer Systeme und Anwendungen. In diversen Bereichen der angewandten Informatik könnten auch Fächer wie Biologie oder Physik von Interesse sein. Spannend ist auch die Kombination mit philologischen Fächern, besonders mit den Sprachwissenschaften oder der philosophischen Logik.

Umgekehrt kann Informatik auch für Studiengänge in allen genannten Fächern als Nebenfach belegt werden.

Spezielle Studiengänge

Einen Schritt weiter als die Wahl eines bestimmten Nebenfaches gehen gezielte Kombinationsstudienfächer. Die wichtigsten von ihnen sind folgende:

- **Wirtschaftsinformatik**. Eine gezielte Kombination aus Themen der (insbesondere angewandten) Informatik und der Betriebswirtschaftslehre. Wirtschaftsinformatiker können in allen Bereichen der kaufmännischen IT-Anwendung arbeiten, vor allem in Banken, Versicherungen oder den Fachabteilungen großer Unternehmen, aber auch in der Entwicklung kaufmännischer Softwarelösungen.
- **Medieninformatik**. Dieser Studiengang stellt den Bereich der multimedialen IT-Anwendung in den Mittelpunkt. Neben den Grundlagen der Informatik lernen die Studierenden vor allem den praktischen IT-Einsatz kennen: Grafik- und Bildbearbeitung, Audio- und Videoschnitt und die Erstellung multimedialer Präsentationen. Von allen IT-Studiengängen erfordert dieser das größte kreative Potential.
- **Bioinformatik**. Die Forschungsergebnisse der modernen Biologie, besonders der Mikrobiologie und der Gentechnik, könnten ohne umfangreiche IT-Unterstützung nicht ausgewertet werden. Bioinformatiker bilden die Schnittstelle zwischen Biologie und Informationstechnik; sie erstellen die Anwendungen zur Analyse biologischer Forschungen, etwa der Sequenzanalyse von Genen. Die Bioinformatik ist möglicherweise der wachstumstärkste Sektor in der gesamten IT-Landschaft.
- **Medizinische Informatik**. Auch in der modernen Medizin spielt Computertechnik eine überaus wichtige Rolle: Von der Verwaltung der Patienten- und Diagnosedaten über die automatisierte Analyse von Laborproben bis hin zum halbautonom arbeitenden OP-Roboter ist der Praxis- und Klinikalltag ohne IT nicht mehr denkbar. In der medizinischen Informatik erwerben die Studierenden sowohl fundierte medizinische als auch wichtige informationstechnische Fachkenntnisse, so dass sie in den verschiedensten Bereichen der Medizintechnik und -praxis arbeiten können.

Studium und dieses Buch

Die Tiefe des für ein Studium erforderlichen Wissens geht weit über das hinaus, was ein einzelnes allgemeines Grundlagenbuch wie das vorliegende leisten kann. Dennoch kann dieses Buch auch im Studium ein unentbehrlicher Begleiter sein: Hier können Sie die Grundlagen der Theorie und Praxis der unterschiedlichen Themen schnell und übersichtlich nachschlagen – als letzte Absicherung vor der entscheidenden Fachklausur oder auch als Anleser für die Veranstaltungsplanung des kommenden Semesters.

1.2 Die Geschichte der Rechenmaschinen und Computer

In diesem Buch ist von Computern die Rede, und natürlich wissen Sie ganz genau, was das ist. Wenn Sie allerdings den Versuch machen sollten, einen Computer mit allen Dimensionen seiner heutigen Möglichkeiten griffig zu definieren – wie würde diese Definition lauten? Vergangene Generationen sprachen etwa von **programmgesteuerten Rechenautomaten**, und gerade Fachleute scheinen noch heute lieber Rechner als Computer zu dieser Maschine zu sagen.

Sind Computer nur Rechner?

Aber ist Rechnen heutzutage die wichtigste Aufgabe der Computer? Es scheinen schließlich mehr Leute einen Webbrowser zu verwenden als die Tabellenkalkulation Excel, und das Bildbearbeitungsprogramm Photoshop ist erheblich populärer als spezielle Mathematikersoftware wie Mathematica oder Maple.

2 Mathematische und technische Grundlagen

Unser Dasein liegt zwischen zwei Ewigkeiten.
– Blaise Pascal

Dieses Kapitel bildet die Grundlage für das Verständnis von Computerprogrammen. Wer die Logik versteht, die den einzelnen Vorgängen in Computersystemen zugrunde liegt, wird geringere Schwierigkeiten haben, mit Software umzugehen. Die konkrete Verwirklichung maßgeblicher logischer Funktionen durch elektronische Bauteile wird hier ebenfalls behandelt. Das Kapitel wird durch eine einfache Prozessorsimulation abgerundet, die sowohl auf dem Papier als auch als Programm auf einem gewöhnlichen PC mit einer einfachen Sprache programmiert werden kann.

Die grundlegenden Arbeitsschritte, die ein Computer ausführt, sind mathematische und logische Operationen: Der Mikroprozessor verknüpft Werte, die ihm ein Programm übergibt oder die er auf Anweisung aus dem Speicher oder von einem Eingabegerät liest, nach verschiedenen Vorschriften. Dazu gehören Grundrechenarten, Vergleiche und Wenn-dann-Beziehungen.

2.1 Einführung in die Logik

Der Begriff **Logik** ist von dem griechischen Wort λογος (Logos) abgeleitet. Die Bedeutung dieses Wortes hat eine lange Geschichte und ist nicht ganz eindeutig. Die Wurzel des Wortes stammt von dem altgriechischen Verb λεγειν (legéin), das für zunächst für »sammeln« oder »auflesen« steht; es ist verwandt mit lateinisch »legere« und deutsch »lesen«.

Aus der Sicht der Informatik ist die wichtigste Form der Logik die **formale Logik**. Ihre klassische Form ist die **Aussagelogik**, die Wissenschaft von der Verknüpfung und Wechselwirkung von Aussagen. Eine **Aussage** ist dabei ein beliebiger Satz, der **eindeutig wahr** oder **eindeutig falsch** ist, dessen Wahrheit also überprüft werden kann. Für die Mathematik wurde die Aussagelogik erst nutzbar, als **Gottlob Frege** 1879 die **Prädikatenlogik** entwickelte, eine mathematisch-formale Schreibweise für Aussagen, auf der einige der weiter unten verwendeten mathematischen Sätze basieren. Typische Formulierungen der Prädikatenlogik sind die bekannten Satzanfänge »Für alle x gilt: ...« (∀ x) oder »Es gibt (mindestens) ein x, für das gilt: ...« (∃ x).

2.1.1 Aussagen

Aussagen sind beispielsweise die folgenden Sätze:

- »Der Kölner Dom ist 157 Meter hoch.«
- »Der Kölner Dom ist 17 Meter hoch.«
- »Ich bin 17 Meter groß.«

Die folgenden Sätze sind dagegen aus verschiedenen Gründen *keine* Aussagen:

- »Der Kölner Dom ist schön.« – eine subjektive Meinungsäußerung, die für den einen wahr und für den anderen falsch sein kann.
- »Ist heute Freitag?« – eine Frage ist keine Aussage; in diesem Fall kann die Antwort eine Aussage sein.
- »Wie geht es dir?« – ebenfalls eine Frage, allerdings ist hier auch die Antwort keinesfalls eine Aussage (sie ergibt wieder eine subjektive Meinungsäußerung).

Mathematische Aussagen

Die Art von Aussagen, die im Zusammenhang mit Informatik und Computern besonders interessant ist, sind die mathematischen Aussagen. Eine **mathematische Aussage** ist ein System, das aus Termen (mathematischen Ausdrücken) besteht. Ein Term ist im einfachsten Fall eine numerische Konstante wie beispielsweise 100 oder –3,25, in komplexeren Fällen eine arithmetische Verknüpfung wie etwa 3 + 5, die sich in einen konkreten Wert auflösen lassen muss. Eine vollständige mathematische Aussage ist ein Vergleich zwischen Termen, der zwei mögliche Formen annehmen kann:

- Die **Gleichung** ist eine wahre Aussage, wenn die beiden verknüpften Terme den gleichen Wert haben.
- Die **Ungleichung** ist dagegen dann eine wahre Aussage, wenn die Werte der beiden verknüpften Terme auf eine vorgegebene Weise unterschiedlich sind.

Beispiele für mathematische Aussagen sind folgende:

- $5 + 6 = 6 + 5$ – eine Gleichung.
- $5 + 6 < 6 + 5$ – eine Ungleichung; die Beziehung lautet in diesem Fall »ist kleiner als«.

Sowohl sprachliche als auch mathematische Aussagen können, wie bereits erwähnt, wahr oder falsch sein. Hier sehen Sie einige Beispiele für wahre und falsche Aussagen:

- »Der Kölner Dom ist 17 Meter hoch.« – falsche sprachliche Aussage.
- »Ein Tag hat 24 Stunden.« – wahre sprachliche Aussage.
- $5 > 7$ – falsche mathematische Aussage (Ungleichung).
- $7 > 5$ – wahre mathematische Aussage (Ungleichung).
- $3 + 4 = 34$ – falsche mathematische Aussage (Gleichung).
- $3 + 4 = 7$ – wahre mathematische Aussage (Gleichung).

2.1.2 Aussageformen

Die in der Mathematik weit verbreiteten Formeln und verallgemeinerten Gleichungen mit Platzhaltern (Variablen oder Unbekannten) sind **keine** Aussagen. Schließlich gilt für sie die Bedingung nicht, dass sie eindeutig wahr oder falsch sein müssen. Beispielsweise lässt sich die allgemeine Gleichung $a + b = 3$ mit unendlich vielen Werten zur wahren oder auch zur falschen Aussage machen:

- $a = 2; b = 1$ oder $a = 3{,}5; b = -0{,}5$ ergeben eine wahre Aussage.
- $a = 3; b = 5$ oder $a = 0{,}75; b = 0{,}2$ ergeben eine falsche Aussage.

Ein Wert, der in eine Aussageform eingesetzt wird und diese zu einer wahren Aussage macht, wird als **Lösung** dieser Aussageform beziehungsweise genauer als Teil ihrer **Lösungsmenge** bezeichnet.

Gleichungen und Ungleichungen lösen

Das folgende Beispiel zeigt, wie die Lösungsmenge einer linearen Gleichung mit einer Unbekannten bestimmt wird:

$$\begin{aligned} 2x + 7 &= 21 \quad | -7 \\ 2x &= 14 \quad | :2 \\ x &= 7 \end{aligned}$$

Lineare Gleichungen besitzen in der Regel genau eine Lösung. Komplexere Gleichungssysteme können dagegen auch Lösungsmengen besitzen, die aus keiner, aus mehreren oder aus unendlich vielen Lösungen bestehen.

Das Lösen von Ungleichungen ist der Versuch, eine Menge von Werten zu bestimmen, die in die Aussageform einer Ungleichung eingesetzt werden können, um eine wahre Aussage zu erhalten:

$$\begin{aligned} 2x + 7 &< 21 \quad | -7 \\ 2x &< 14 \quad | :2 \\ x &< 7 \end{aligned}$$

Die Lösungsmenge der Ungleichung ist die Menge aller x, für die gilt, dass x kleiner als 7 ist. Mathematisch wird dies folgendermaßen ausgedrückt:

$$L = \{ x \mid x < 7 \}$$

2.1.3 Logische Verknüpfungen

Nicht immer werden nur einzelne Aussagen betrachtet. Es ist üblich und oft notwendig, mehrere Aussagen mit Hilfe verschiedener logischer Verknüpfungen zu verbinden. Dazu gehören sowohl logische Schlussfolgerungen (Wenn-dann-Beziehungen) als auch einfache Und- sowie Oder-Verknüpfungen.

Boolesche Algebra

Das gesamte Fachgebiet der logischen Verknüpfungen wird auch als boolesche Algebra bezeichnet, benannt nach dem britischen Mathematiker **George Boole**, der bereits im 19. Jahrhundert eine Algebra der binären logischen Verknüpfungen entwickelte.

Logische Schlussfolgerungen

Die Schlussfolgerung, das Lieblingsspielzeug der philosophischen Logik, ist eine Wenn-dann-Beziehung zwischen Aussagen. Auch Computerprogramme arbeiten mit dieser Art der Logik, weil sie oft anhand bestimmter Bedingungen Entscheidungen treffen müssen.

In der sprachlichen Aussagelogik sieht eine Schlussfolgerung beispielsweise folgendermaßen aus:

1. Es regnet.
2. Die Straße wird nass.

Schlussfolgerung: Wenn es regnet, dann wird die Straße nass.

Oder in formaler Schreibweise: Es regnet. ⇒ Die Straße wird nass. Das Zeichen ⇒ wird als »daraus folgt« aufgelöst.

Die direkte Umkehrung einer solchen Schlussfolgerung ist unzulässig: Eine Aussage wie

$A \Rightarrow B$ (aus A folgt B)

bedingt nicht den folgenden Schluss:

$B \Rightarrow A$ (aus B folgt A)

Sprachlich formuliert gilt also nicht die folgende Schlussfolgerung: Wenn die Straße nass wird, regnet es. Schließlich kann es zahlreiche Gründe dafür geben, warum eine Straße nass wird.

Umkehrschluss

Den korrekten, zulässigen Umkehrschluss bildet dagegen die verneinte Umkehrung:

$\neg B \Rightarrow \neg A$ (aus nicht-B folgt nicht-A)

Auf das Sprachbeispiel übertragen: Wenn die Straße nicht nass wird, dann regnet es nicht. Dieser Satz ist intuitiv einleuchtend und richtig.

Zur Verdeutlichung sehen Sie hier alle erlaubten Umkehrschlüsse im Überblick:

1. $A \Rightarrow B$; Umkehrung: $\neg B \Rightarrow \neg A$

   Beispiel: Wenn es regnet, dann wird die Straße nass. Umkehrung: Wenn die Straße nicht nass wird, dann regnet es nicht.

2. $A \Rightarrow \neg B$; Umkehrung: $B \Rightarrow \neg A$

   Beispiel: Wenn die Sonne scheint, dann ist nicht Nacht. Umkehrung: Wenn es Nacht ist, dann scheint nicht die Sonne.[1]

3. $\neg A \Rightarrow B$; Umkehrung: $\neg B \Rightarrow A$

   Beispiel: Wenn nicht Wochenende, Feiertag oder Urlaub ist, dann muss ich arbeiten. Umkehrung: Wenn ich nicht arbeiten muss, dann ist Wochenende, Feiertag oder Urlaub.

4. $\neg A \Rightarrow \neg B$; Umkehrung: $B \Rightarrow A$

   Beispiel: Wenn ich nicht den Führerschein habe, dann darf ich nicht Auto fahren. Umkehrung: Wenn ich Auto fahren darf, dann habe ich den Führerschein.

Und- und Oder-Verknüpfungen

Die beiden Verknüpfungen Und und Oder (AND und OR) sind die wichtigsten Mittel zur Verbindung mehrerer Aussagen oder Bedingungen.

Und-Verknüpfung

Werden mehrere Aussagen durch Und verknüpft (Konjunktion), ergibt sich nur dann eine wahre Aussage, wenn alle Teilaussagen wahr sind.

1 Natürlich gibt es Gegenden auf der Welt, wo dieser Bezug nicht zutrifft. Wie bei den meisten Aussagen gibt es Nebenbedingungen, die stillschweigend gelten – hier beispielweise: »Wenn die Sonne in Mitteleuropa scheint, dann ist nicht Nacht.«

Die Verknüpfung A ∧ B ist also nur dann wahr, wenn sowohl A als auch B wahre Aussagen sind.

Sprachlich kann dies sehr gut durch zusammengesetzte Wenn-Bedingungen bei der Schlussfolgerung gezeigt werden. Zum Beispiel:

Wenn die Glühbirne funktioniert **und** der Lichtschalter eingeschaltet wird, dann leuchtet das Licht. Oder als einzelne Aussagen:

1. Die Glühbirne funktioniert.
2. Der Lichtschalter ist eingeschaltet.
3. Das Licht leuchtet.

Die erste und die zweite Aussage werden hier durch Und verknüpft und ergeben so die dritte Aussage:

Die Glühbirne funktioniert ∧ der Lichtschalter wird eingeschaltet ⇒ das Licht leuchtet.

Schematisch lassen sich solche Verknüpfungen am einfachsten durch eine so genannte Wahrheitstabelle darstellen. In einer solchen Tabelle (wie übrigens auch bei den Abläufen innerhalb eines Computers) wird eine wahre Aussage durch den Wert 1 und eine falsche Aussage durch die 0 dargestellt. Die Wahrheitstabelle für Und sieht also folgendermaßen aus:

∧	**0**	**1**
0	0	0
1	0	1

Tabelle 2.1 Wahrheitstabelle für die Und-Verknüpfung

Oder-Verknüpfung

Werden dagegen mehrere Aussagen durch Oder verknüpft (Disjunktion), dann ist die Gesamtaussage wahr, wenn mindestens eine der Teilaussagen zutrifft.

Die Beziehung A ∨ B ist also wahr, wenn A wahr ist, wenn B wahr ist, oder auch, wenn A und B wahr sind. Hier sehen Sie zunächst wieder ein sprachliches Beispiel:

Wenn ich eine Million im Lotto **oder** bei einer Quiz-Sendung gewinne, dann bin ich Millionär. Die einzelnen Aussagen sind folgende:

1. Ich gewinne eine Million im Lotto.
2. Ich gewinne eine Million bei einer Quiz-Sendung.
3. Ich bin Millionär.

Formal sieht diese Oder-Verknüpfung der beiden ersten Aussagen so aus:

Ich gewinne eine Million im Lotto ∨ ich gewinne bei einer Quiz-Sendung ⇒ ich bin Millionär. Es ist egal, ob nur eine der beiden Voraussetzungen zutrifft oder ob beide wahr sind – in beiden Fällen stimmt die Schlussfolgerung.

Hier wieder die entsprechende schematische Darstellung als Wahrheitstabelle.

∨	**0**	**1**
0	0	1
1	1	1

Tabelle 2.2 Wahrheitstabelle für die Oder-Verknüpfung

Die folgende Tabelle zeigt eine Übersicht der Und- und Oder-Verknüpfungen aller vier möglichen logischen Belegungen zweier Wahrheitswerte A und B sowie all ihrer Verneinungen.

Verknüpfung	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1
A ∧ B	0	0	0	1
A ∨ B	0	1	1	1
¬A ∧ B	0	1	0	0
¬A ∨ B	1	1	0	1
A ∧ ¬B	0	0	1	0
A ∨ ¬B	1	0	1	1
¬ (A ∧ B)	1	1	1	0
¬ (A ∧ B)	1	0	0	0

Tabelle 2.3 Wahrheitstabelle aller normalen und verneinten Und- beziehungsweise Oder-Verknüpfungen

Aus den Verhältnissen in Tabelle 2.3 ergeben sich die beiden folgenden wichtigen Äquivalenzen, die als **DeMorgan-Theorem** bezeichnet werden:

$$\neg A \wedge \neg B \Leftrightarrow \neg (A \vee B)$$

$$\neg A \vee \neg B \Leftrightarrow \neg (A \wedge B)$$

Diese Beziehungen müssen Sie direkt beim Programmieren beachten, wenn Sie verknüpfte Wenn-dann-Bedingungen formulieren.

XOR-Verknüpfung

Zu guter Letzt gibt es eine dritte wichtige logische Verknüpfung, die als **Exklusiv-Oder** bezeichnet wird. Da sie in der Mathematik nicht verwendet wird, besitzt sie kein traditionelles Zeichen, sondern wird in der Regel durch den abgekürzten englischen Namen XOR bezeichnet. Auch in den meisten Programmiersprachen existiert kein XOR-Operator.

Eine XOR-Verknüpfung zweier Aussagen ist nur dann wahr, wenn genau eine Teilaussage wahr ist. Es ist recht schwierig, ein anschauliches sprachliches Beispiel für diese Verknüpfung zu finden. Das XOR steht im Grunde für ein »Entweder-Oder«: Wenn ich entweder mit dem Bus oder mit dem Auto fahre, gelange ich zur Arbeit (ich kann auf keinen Fall mit Bus und Auto gleichzeitig fahren).

Tabelle 2.4 zeigt die möglichen Ergebnisse der XOR-Verknüpfung.

XOR	0	1
0	0	1
1	1	0

Tabelle 2.4 Wahrheitstabelle für die XOR-Verknüpfung

Vergleichsoperationen

Diese Art logischer Verknüpfungen erzeugt Aussagen durch die Überprüfung von Termen auf Gleichheit oder Ungleichheit. Die resultierenden Aussagen können also wahr oder falsch sein.

- Gleichheit:

  $A = B$ (A gleich B) ist wahr, wenn A den gleichen Wert hat wie B.
- Ungleichheit:

  $A \neq B$ (A ungleich B) ist wahr, wenn A einen anderen Wert hat als B.
- Ungleichheit mit Richtungsangabe:
  - $A < B$ (A ist kleiner als B) ist wahr, wenn A einen geringeren Wert hat als B.
  - $A > B$ (A ist größer als B) ist wahr, wenn A einen höheren Wert hat als B.
  - $A \leq B$ (A ist kleiner oder gleich B) ist wahr, wenn A entweder einen geringeren Wert als B oder den gleichen Wert wie B hat.
  - $A \geq B$ (A ist größer oder gleich B) ist wahr, wenn A entweder einen höheren Wert als B oder den gleichen Wert wie B hat.
  - $A \leq B$ ist eine Kurzfassung für: $A < B \vee A = B$
  - $A \geq B$ ist entsprechend eine Kurzfassung für: $A > B \vee A = B$

Umgekehrte Vergleichsoperationen

Interessant sind einige Umkehrungen (gegenteilige Aussagen) der Vergleichsoperationen:

- Die Umkehraussage von $A = B$ ist $A \neq B$.
- $A < B$ besitzt die Umkehraussage $A \geq B$.
- $A > B$ hat schließlich die Umkehraussage $A \leq B$.

Tabelle 2.5 verdeutlicht dies an Beispielen für den Vergleich verschiedener ganzzahliger Werte. Beachten Sie, dass die verknüpften Werte ganze Zahlen, die Ergebnisse (1 oder 0) aber Wahrheitswerte sind.

Verknüpfung	A=2, B=2	A=2, B=3	A=3, B=2
A = B	1	0	0
A ≠ B	0	1	1
A < B	0	1	0

Tabelle 2.5 Beispiele für Vergleichsoperationen

Verknüpfung	A=2, B=2	A=2, B=3	A=3, B=2
A > B	0	0	1
A ≤ B	1	1	0
A ≥ B	1	0	1

Tabelle 2.5 Beispiele für Vergleichsoperationen (Forts.)

Logische Verknüpfungen in Computerprogrammen

Da Zeichen wie ≠, ≤ oder ∨ nicht zum ASCII-Zeichensatz (dem ursprünglichen Standard-Computerzeichensatz) gehören, haben sich die Entwickler der verschiedenen Programmiersprachen andere Zeichen beziehungsweise Zeichenkombinationen ausgedacht. Ärgerlicherweise sind diese Bezeichnungen in den verschiedenen Sprachen nicht einheitlich. Tabelle 2.6 zeigt die Schreibweisen der Programmiersprachen C und Pascal im Vergleich. Beachten Sie, dass die C-Schreibweise in einer ganzen Reihe von Sprachen verwendet wird, beispielsweise C++, Java, JavaScript, Perl, PHP und so weiter. Die Pascal-Schreibweise verwenden dagegen auch einige BASIC-Dialekte.

Mathematik	C-Schreibweise	Pascal-Schreibweise
a = b (Vergleich)	a == b	a = b
a = b (Wertzuweisung)	a = b	a := b
a ≠ b	a != b	a <> b
a < b	a < b	a < b
a > b	a > b	a > b
a ≤ b	a <= b	a <= b
a ≥ b	a >= b	a >= b
a ∧ b	a && b	a AND b
a ∨ b	a || b	a OR b

Tabelle 2.6 Schreibweise logischer Operationen in verschiedenen Programmiersprachen

Die in der Tabelle aufgeführte Wertzuweisung betrifft eine Besonderheit von Variablen in Programmiersprachen: Während eine Variable in der Mathematik lediglich ein Platzhalter ist, der für beliebige Werte stehen kann, ist sie in einer Programmiersprache ein Speicherplatz, der jederzeit einen bestimmten, wohl definierten Wert enthält. Die Wertzuweisungsoperation dient dazu, ihr einen solchen Wert zuzuteilen.

2.1.4 Mengenoperationen

Eine spezielle Form der logischen Verknüpfung beschäftigt sich mit den Beziehungen zwischen einem einzelnen Element und einer Menge beziehungsweise zwischen zwei Men-

gen. Eine Menge ist eine Gruppe mehrerer Werte, die entweder als Abfolge einzelner Zahlen oder durch bestimmte Regeln definiert wird.

Beachten Sie für die Umsetzung im Computer, dass Mengenoperationen nicht in jeder Programmiersprache eingebaut sind. In Kapitel 6, *Konzepte der Programmierung*, werden allerdings einige Strategien vorgestellt, wie man Listen oder Mengen in verschiedenen Sprachen erzeugen und damit arbeiten kann.

Beziehungen zwischen Mengen und einzelnen Werten

Ein Wert ist **Element** einer Menge, wenn dieser Wert in der Menge vorkommt. Ein Wert ist **nicht Element** einer Menge, wenn dieser Wert nicht in der Menge vorkommt.

Betrachten Sie zum Beispiel die Menge M, für die die folgende Definition gilt:

$M := \{3; 4; 5\}$

Es gelten die folgenden Elementbeziehungen:

- ist Element von M. Formale Schreibweise: $3 \in M$
- ist nicht Element von M. Formale Schreibweise: $2 \notin M$.

Eine weitere Menge P sei folgendermaßen definiert:

$P := \{x | x < 5 \land x \in \mathbb{R}\}$

P ist also die Menge aller x, für die gilt: x ist kleiner als 5 und x ist Element der Menge der reellen Zahlen. Für diese Menge gelten die folgenden Elementbeziehungen:

$4 \in P$, aber $5 \notin P$.

Beziehungen zwischen zwei Mengen

Eine Menge M heißt **Teilmenge** einer Menge N, wenn jedes Element von M auch Element von N ist, wenn also für jedes $x \in M$ auch $x \in N$ gilt.

Eine Menge M ist **nicht Teilmenge** einer Menge N, wenn es in M mindestens ein Element gibt, das nicht auch Element von N ist. Es gibt also mindestens ein $x \in M$, für das gilt: $x \notin N$.

Betrachten Sie beispielsweise die folgenden Mengendefinitionen:

$M := \{3; 4; 5; 6\}$
$N := \{2; 4; 5; 6\}$
$P := \{3; 4; 5; 6; 7\}$

Es gelten die folgenden Mengenbeziehungen:

$M \subset P$ (M ist Teilmenge von P)

$N \not\subset P$ (N ist nicht Teilmenge von P)

Echte Teilmengen

Übrigens ist die angegebene Teilmengendefinition ungenau: Die oben beschriebene Beziehung könnte ebenso gut bedeuten, dass zwei identische Mengen beschrieben werden.

Deshalb heißt eine Menge M **echte Teilmenge** einer Menge N, wenn folgende Bedingungen gelten:

- Für jedes $x \in M$ gilt auch $x \in N$.
- Es existiert mindestens ein $x \in N$, für das $x \notin M$ gilt.

Wenn man es genau nimmt, wird nur für diese echte Teilmenge die Schreibweise $M \subset N$ verwendet. Für die weiter oben definierte allgemeine Teilmenge, bei der $M = N$ als Variante zulässig ist, wird stattdessen die Formulierung $M \subseteq N$ (Teilmenge oder gleich) verwendet.

Wenn $M \subseteq N$ gilt, wird N übrigens umgekehrt als **Obermenge** von M bezeichnet. Geschrieben wird dies als $N \supseteq M$ (N ist Obermenge von oder gleich M). Die strengere Form $M \subset N$ (echte Teilmenge) bedeutet entsprechend, dass N **echte Obermenge** von M ist: $N \supset M$.

Eine Abfolge von Beziehungen echter Teilmengen beziehungsweise Obermengen lässt sich hervorragend an den offiziellen Zahlenmengen der Mathematik demonstrieren. Dies sind im Einzelnen (von der speziellsten bis zur allgemeinsten) folgende:

1. **Die Menge der natürlichen Zahlen.** Natürliche Zahlen sind alle Zahlen, mit denen sich Anzahlen ausdrücken lassen. Intuitiv ist diese Menge folgendermaßen definiert:

   $\mathbb{N} = \{1; 2; 3; 4; ...\}$

   Die 0 gehört übrigens nicht zur Menge der natürlichen Zahlen, allerdings gibt es die spezielle Menge N_0, die zusätzlich die 0 enthält.
2. **Die Menge der ganzen Zahlen.** Ebenso wie die natürlichen Zahlen sind auch die ganzen Zahlen intuitiv betrachtet Zahlen ohne Nachkommastellen, allerdings mitsamt 0 und negativen Zahlen. Es handelt sich um die folgende Menge:

   $\mathbb{Z} = \{...; -3; -2; -1; 0; 1; 2; 3; ...\}$

3. **Die Menge der rationalen Zahlen.** Es handelt sich um sämtliche Zahlen, die durch die Division zweier ganzer Zahlen gebildet werden können. Dies sind neben den ganzen Zahlen selbst alle abbrechenden und alle periodischen Dezimalbrüche. Formal lautet die Definition dieser Menge folgendermaßen:

   $\mathbb{Q} = \{x \mid x = {}^{p}/_{q} \wedge p \in \mathbb{Z} \wedge q \in \mathbb{Z} \wedge q \neq 0\}$

   Bemerkenswert an dieser Menge ist, dass zwischen zwei rationalen Zahlen immer unendlich viele weitere rationale Zahlen liegen. Bei den natürlichen und ganzen Zahlen ist das anders: Da zwei Elemente dieser Mengen jeweils einen festgelegten numerischen Abstand voneinander haben (1), ist die Anzahl der Elemente zwischen zweien von ihnen jeweils endlich und steht fest.
4. **Die Menge der reellen Zahlen.** Neben den abbrechenden und den periodischen Dezimalbrüchen gibt es auch unendlich viele nichtabbrechende, nichtperiodische. Es handelt sich also um Zahlen mit unendlich vielen Nachkommastellen ohne Wiederholung (Periode). Beispiele sind etwa die Kreiszahl π (3,1415926…), die Eulersche Zahl e (2,718281828…) – die Basis des natürlichen Logarithmus – oder $\sqrt{2}$ (1,41421356…).

Formal haben diese Zahlen miteinander gemeinsam, dass ihr Quadrat eine positive Zahl oder 0 ist:

$\mathbb{R} := \{ x \mid x^2 \geq 0 \}$

Beachten Sie, dass reelle Zahlen im Computer nicht dargestellt werden können, übrigens ebenso wenig wie periodische Dezimalbrüche. Wie weiter unten ausgeführt, verwenden Rechner eine bestimmte Anzahl von Bits und können Fließkommazahlen auf diese Weise mit einer bestimmten Genauigkeit, also nur mit einer endlichen Anzahl von Stellen, speichern.

5. **Die Menge der komplexen Zahlen.** Eine Zahl mit negativem Quadrat ist intuitiv nicht vorstellbar (da sowohl positive als auch negative Zahlen, wenn man sie mit sich selbst multipliziert, zu einem positiven Ergebnis führen). Dennoch ist es zum Beispiel für mathematische Gedankenexperimente oder bestimmte Berechnungen aus der theoretischen Physik erforderlich, die Wurzeln negativer Zahlen zu berechnen.

   Zu diesem Zweck werden die **imaginären** (oder irreellen) **Zahlen** eingeführt. Sie reduzieren das Problem der Wurzel aus negativen Zahlen auf die irrationale Komponente *i*, die als $\sqrt{-1}$ definiert ist (und natürlich nicht durch eine normale Berechnung aufgelöst werden kann). Jede irreelle Zahl ist deshalb als Vielfaches von *i* definiert.

   Die reellen und die irreellen Zahlen werden zusammengenommen als komplexe Zahlen ($\mathbb{C}$) bezeichnet und bilden die vollständigste Zahlenmenge der allgemeinen Mathematik.

Jede der fünf hier genannten Mengen enthält die vorige. Die verkettete Beziehung von Teil- beziehungsweise Obermengen lautet also:

$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$

$\mathbb{C} \supset \mathbb{R} \supset \mathbb{Q} \supset \mathbb{Z} \supset \mathbb{N}$

Verknüpfungen von Mengen

Ähnlich, wie Sie einzelne Werte durch arithmetische Operatoren oder durch logische Verknüpfungen miteinander verbinden können, existieren spezielle Mengenoperationen, deren Ergebnis eine Verknüpfung der ursprünglichen Mengen ist.

- **Die Schnittmenge.** Eine Schnittmenge $M \cap N$ zweier Mengen M und N enthält alle diejenigen Elemente x, für die $x \in M$ und $x \in N$ gilt. Hier sehen Sie ein Beispiel:

  $M := \{1; 2; 3; 4\}$
  $N := \{4; 5; 6; 7\}$
  $M \cap N = \{4\}$

  Beachten Sie, dass das Ergebnis nicht 4 lautet, sondern »die Menge, in der nur die 4 enthalten ist«. Eine Schnittmenge ist also auch dann eine Menge, wenn sie nur ein Element enthält. Wenn die beiden verknüpften Mengen keine gemeinsamen Elemente besitzen, geschieht sogar Folgendes:

$M := \{1; 2; 3\}$
$N := \{4; 5; 6\}$
$M \cap N = \{\ \} = \varnothing$

{ } beziehungsweise ∅ wird dabei als leere Menge bezeichnet.

- **Die Vereinigungsmenge.** Eine Vereinigungsmenge $M \cup N$ zweier Mengen M und N ist die verbundene Menge aller $x \in M$ und aller $y \in N$. Hier ein Beispiel:

  $M := \{1; 2; 3; 4\}$
  $N := \{4; 5; 6; 7\}$
  $M \cup N = \{1; 2; 3; 4; 5; 6; 7\}$

- **Die Restmenge.** Wenn aus der Menge M alle Elemente einer Menge N entfernt werden ($M \setminus N$, gesprochen »M ohne N«), dann ist das Ergebnis eine Restmenge. Beispiel:

  $M := \{1; 2; 3; 4\}$
  $N := \{3; 4; 5\}$
  $M \setminus N = \{1; 2\}$

2.2 Informationsspeicherung im Computer

Wie Sie bereits in der Einführung erfahren haben und wahrscheinlich schon vorher wussten, speichern Computer Informationen als Abfolge von Einsen und Nullen ab. Diese Darstellungsform wird als **binäre Codierung** bezeichnet. Die duale Darstellung ganzer Zahlen im entsprechenden Zahlensystem, nämlich dem dualen oder Zweiersystem, ist nur ein Sonderfall der binären Darstellung.

2.2.1 Bits und Bytes

Eine einzelne Binärstelle, die ein Rechner speichert, wird als **Bit** bezeichnet. Das ist die Abkürzung für »BInary digiT«, also Binärziffer. Es handelt sich um die kleinste Informationseinheit, die ein Computer verarbeiten kann – gleichzeitig ist es auch die kleinste vorstellbare Informationseinheit. Um überhaupt Daten darstellen zu können, benötigt man einen Zeichencode oder ein »Alphabet« aus mindestens zwei Elementen.

Weniger Information als 1 Bit?

Natürlich könnte man Zahlen oder andere Daten auch einfach durch »Strichlisten« darstellen – eine bestimmte Anzahl des immer gleichen Informationselements steht für das jeweilige Symbol. Aber spätestens zur Darstellung der Lücke zwischen zwei auf diese Weise codierten Symbolen wäre dann doch wieder eine zweite Informationsart erforderlich.

Zahlensysteme

Für das Verständnis der Speicherung von Werten im Computer benötigen Sie Kenntnisse der verschiedenen Zahlensysteme. Bereits in Kapitel 1, *Einleitung*, wurde die Bedeutung der Stellenwertsysteme als Voraussetzung für die Erfindung von Rechenmaschinen und Computern hervorgehoben.

Stellenwertsysteme

In einem Stellenwertsystem hängt der Wert einer einzelnen Ziffer von zwei Faktoren ab: ihrem Eigenwert und ihrer Position in der Zahl. Jede Stelle besitzt einen festen Grundwert,

6 Konzepte der Programmierung

Intelligente Fehler zu machen ist eine große Kunst.
– Federico Fellini

Nachdem Sie im vorigen Kapitel die grundlegende Syntax der Programmierung in drei verschiedenen Sprachen erlernt haben, ist es nun Zeit, dieses Handwerkszeug in der Praxis einzusetzen. In diesem Kapitel werden deshalb verschiedene konkrete Aspekte der Programmierung vorgestellt, von der Implementierung wichtiger Algorithmen und Datenstrukturen über eine Einführung in die systemnahe Programmierung bis hin zur GUI-Anwendungsentwicklung und der Arbeit mit einer integrierten Entwicklungsumgebung.

Die Implementierung der verschiedenen Beispiele erfolgt jeweils in einer der drei im vorigen Kapitel vorgestellten Programmiersprachen; es wird für jede Aufgabe eine passende Sprache ausgewählt.

6.1 Algorithmen und Datenstrukturen

Wenn Sie Programmieraufgaben lösen müssen, genügt es nicht, die einzelnen Befehle und Datentypen der gewählten Programmiersprache zu kennen. Sie benötigen vor allem ein Gespür dafür, wie sich Informationen und Tätigkeiten aus der realen Welt am effektivsten in einem Computerprogramm darstellen lassen. Für die Speicherung der verschiedenen Arten von Informationen werden mit Hilfe der Datentypen einer Programmiersprache so genannte **Datenstrukturen** entwickelt. Die Strategien zur Verarbeitung der Datenstrukturen werden als **Algorithmen** bezeichnet – der theoretische Hintergrund der Algorithmik wurde bereits in Kapitel 2, *Mathematische und technische Grundlagen*, angesprochen. In diesem Abschnitt werden einige verbreitete Standard-Algorithmen und Datenstrukturen dargestellt.

6.1.1 Ein einfaches Praxisbeispiel

Die Entwicklung eines Algorithmus soll hier zunächst Schritt für Schritt an einem Beispiel erläutert werden. Bevor Sie nämlich konkreten Programmcode hinschreiben, sollten Sie sich zunächst mit Papier und Bleistift theoretische Gedanken über den Ablauf des gewünschten Programms machen. Das Ergebnis solcher Überlegungen ist ein Flussdiagramm.

Es soll ein Algorithmus entwickelt werden, um den **größten gemeinsamen Teiler** (GGT) zweier ganzzahliger Werte zu ermitteln. Dies ist eine der wichtigsten Aufgaben im Bereich der Bruchrechnung. Der GGT zweier Werte *m* und *n* ist im günstigsten Fall *m/2* oder *n/2*, im ungünstigsten Fall 1. Wenn er den Wert 1 hat, werden die beiden Zahlen als **teilerfremd** bezeichnet. Abbildung 6.1 zeigt einen Ansatz für ein Flussdiagramm, das eine mögliche Lösung beschreibt.

Es gibt erheblich effizientere Verfahren zur GGT-Berechnung. Hier geht es aber nur um das grundlegende Prinzip.

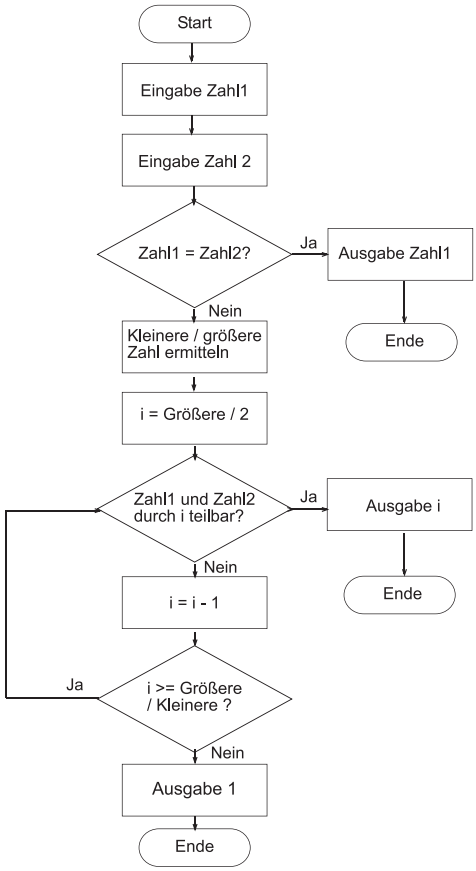

Abbildung 6.1 Flussdiagramm für einen Algorithmus zur Ermittlung des GGT

Im Grunde werden also Schritt für Schritt alle in Frage kommenden Werte ausprobiert. Begonnen wird beim Quotienten der größeren Zahl. Hier sehen Sie eine Implementierung dieses Algorithmus in Java:

```
static int ggt(int m, int n) {
   // Trivialer Fall: m und n gleich:
   if (m == n)
      return m;
   // Größeren und kleineren Wert ermitteln:
   int gr = (m > n) ? m : n;
   int kl = (m < n) ? m : n;
   // Jeden möglichen Teiler testen:
   for (int i = gr / 2; i >= gr / kl; i--) {
      // Sind beide Werte durch i teilbar?
```

```
      if (gr % i == 0 && kl % i == 0)
         return i;
   }
   // Hier bleibt nur noch die 1:
   return 1;
}
```

Da die Methode `ggt()` als `static` deklariert wurde, kann sie innerhalb einer Programmklasse von der Methode `main()` aus aufgerufen werden, ohne dass ein Objekt der Klasse bestehen muss. `static`-Methoden sind gewissermaßen der Java-Ersatz für imperative Funktionen. Hier sehen Sie ein Beispiel für eine `main()`-Methode, die den GGT zweier Kommandozeilenparameter ausgibt:

```
public static void main (String args[]) {
   int m = Integer.parseInt (args[0]);
   int n = Integer.parseInt (args[1]);
   System.out.println ("Der GGT von " + m + " und "
        + n + " ist " + ggt(m, n));
}
```

Nach dem Kompilieren können Sie das Programm beispielsweise folgendermaßen aufrufen, falls die Klasse `GGTtest` heißt:

```
$ java GGTtest 18 24
Der GGT von 18 und 24 ist 6.
```

Die statische Methode `parseInt()` der Klasse `Integer` (einer objektorientierten Umhüllung für einen `int`-Wert) wandelt den übergebenen String in eine Ganzzahl um.

6.1.2 Sortier-Algorithmen

Zu den häufigsten Aufgaben bei der Programmierung gehört die Sortierung von Daten nach einem bestimmten Kriterium. Im Laufe der Informatikgeschichte wurden Dutzende von verschiedenen Sortier-Algorithmen entwickelt, die für unterschiedliche Einsatzzwecke geeignet sind. In diesem Unterabschnitt werden zwei davon vorgestellt. Der erste ist intuitiv verständlich, der zweite gerade für größere Datenmengen besonders effizient.

BubbleSort

Dieser einfachste aller Sortier-Algorithmen durchwandert ein Array elementweise: Er vergleicht jedes Element mit seinem direkten Nachbarn. Wenn die beiden Nachbarn in der falschen Reihenfolge stehen, ist es erforderlich, sie zu vertauschen. Außerdem steht dann fest, dass das Array noch nicht fertig sortiert ist, so dass ein weiterer Durchgang erforderlich ist.

BubbleSort in Perl

Hier sehen Sie eine Implementierung von BubbleSort als Perl-Subroutine, die eine Listenreferenz entgegennimmt und die zugrunde liegende Liste nach diesem Schema sortiert:

```
sub bubblesort {
   $listref = shift;   # Listenreferenz entgegennehmen
```

```
    @liste = @$listref; # Kopie der Liste anfertigen
    do {
       $sortiert = 1;   # Annahme: Bereits sortiert.
       for ($i = 0; $i < $#liste; $i++) {
          # Element i größer als Element i + 1?
          if ($liste[$i] > $liste[$i + 1]) {
             # Die beiden Elemente vertauschen:
             ($liste[$i], $liste[$i + 1])
                  = ($liste[$i + 1], $liste[$i]);
             # Aha! Noch nicht sortiert:
             $sortiert = 0;
          }
       }
    } while (!$sortiert);
    return @liste;      # Sortiert zurückgeben
}
```

Das Kopieren der Liste und die Rückgabe der Kopie sind nicht unbedingt erforderlich, verhindern aber das umständliche Hantieren mit Dereferenzierungen im Kern des eigentlichen Algorithmus. Wenn am Anfang nicht die Kopie `@liste` angefertigt würde, müsste zum Beispiel der eigentliche Tauschbefehl folgendermaßen lauten:

```
($$listref[$i], $$listref[$i + 1])
    = ($$listref[$i + 1], $$listref[$i]);
```

Die spezielle Formulierung `$#array` gibt übrigens den höchsten Index des Arrays `@array` zurück, bei einem Array mit 13 Elementen also beispielsweise 12. Da für den Vergleich jedes Elements mit seinem rechten Nachbarn eine Schleife vom ersten bis zum vorletzten Element benötigt wird, lautet die Bedingung der `for()`-Schleife im vorliegenden Fall `$i < $#liste`.

Die Subroutine `bubblesort` können Sie zum Beispiel so aufrufen:

```
@zahlen = qw(2 7 9 1 3 8 6 7);
@zahlen = bubblesort (\@zahlen);
print join (", ", @zahlen);
```

Die Ausgabe dieses Beispiels lautet:

```
1, 2, 3, 6, 7, 7, 8, 9
```

Beachten Sie, dass für die Sortierung von Strings eine separate Fassung erforderlich wäre, die den Vergleichsoperator `gt` statt des numerischen `>` verwendet (siehe voriges Kapitel). Wenn Sie im Übrigen nicht aufsteigend, sondern absteigend sortieren möchten, muss statt `>` der Operator `<` eingesetzt werden, bei Strings entsprechend `lt`.

BubbleSort in Java

Zu Vergleichszwecken sehen Sie hier eine weitere Implementierung von BubbleSort als Java-Methode, die ein Array von `int`-Werten sortiert:

```
static int[] bubbleSort (int[] liste) {
   boolean sortiert;
```

```
    do {
       sortiert = true;
       for (int i = 0; i < liste.length - 1; i++) {
          if (liste[i] > liste[i + 1]) {
             // Tauschen:
             int temp = liste[i];
             liste[i] = liste[i + 1];
             liste[i + 1] = temp;
             // Nicht sortiert!
             sortiert = false;
          }
       }
    } while (!sortiert);
    return liste;
 }
```


Die folgende `main()`-Methode generiert ein Array mit unsortierten Werten, ruft `bubbleSort()` auf und gibt anschließend die sortierte Liste aus:

```
public static void main (String args[]) {
   int[] werte = {3, 7, 1, 9, 2, 5, 2};
   werte = bubbleSort (werte);
   for (int i = 0; i < werte.length; i++) {
      System.out.print (werte[i] + " ");
   }
}
```

Die Ausgabe sieht natürlich folgendermaßen aus:

```
1 2 2 3 5 7 9
```

Rekursion

Im nächsten Unterabschnitt wird ein sehr komplexer, aber erheblich effizienterer Sortier-Algorithmus vorgestellt. Dieser verwendet die besondere Technik der **Rekursion**, bei der eine Funktion sich selbst mit geänderten Parametern aufruft.

Rekursion ist immer dann nützlich, wenn Sie gleichartige, ineinander verschachtelte Aufgaben lösen müssen.

Beispielsweise lässt sich die **Fakultät** eines Werts sehr leicht rekursiv berechnen. Die Fakultät *n!* einer natürlichen Zahl *n* ist das Ergebnis der Multiplikation *1 x 2 x 3 x ... x n*. Durch Rekursion lässt sich das Problem schrittweise reduzieren, beispielsweise gilt im ersten Schritt: *n! = n * (n – 1)!*.

Das folgende Beispiel zeigt eine rekursive C-Funktion, die auf diese Weise die Fakultät eines übergebenen Werts berechnet:

```
int fakultaet (int n) {
   if (i > 1)
      return n * fakultaet (n - 1);
   else
      return 1;
}
```

Solange das übergebene `n` noch größer als 1 ist, wird das Problem an einen weiteren Aufruf von `fakultaet()` delegiert; das aktuelle `n` wird mit dem Rückgabewert dieses Aufrufs multipliziert. Die Aufrufe werden schließlich von innen nach außern abgearbeitet: Der letzte oder innerste Aufruf gibt 1 zurück, die mit der 2 des darüber liegenden Aufrufs multipliziert wird und so weiter.

Der Gegenbegriff zur Rekursion ist übrigens **Iteration**. Natürlich lässt sich die Fakultät auch iterativ (in einer Schleife) berechnen, allerdings ist der Aufwand größer und die Schreibweise umständlicher:

```
int fakultaet (int n) {
   int i;
   int fak = 1;
   for (i = 2; i <= n; i++) {
      fak *= i;
   }
   return fak;
}
```

QuickSort

Wie der Name vermuten lässt, handelt es sich bei QuickSort um einen besonders geschwindigkeitsoptimierten Sortier-Algorithmus. Für eine ausreichend große Anzahl von Elementen ist er tatsächlich eines der schnellsten verfügbaren Sortierverfahren. Bei einer zu kleinen Elementanzahl kann er seinen theoretischen Geschwindigkeitsvorteil wegen der relativ aufwändigen Initialisierung nicht ausspielen.

QuickSort-Funktionsweise

QuickSort funktioniert nach dem Verfahren »Teile und herrsche«: Das gesamte zu sortierende Array wird in zwei Teile unterteilt (partitioniert), die an rekursive Aufrufe der Funktion übergeben werden. Dazu wird zunächst ein Vergleichselement namens Median gesucht; die Daten rechts des Medians müssen größer sein, die Daten links von ihm kleiner. Dies geschieht für jede Partition so lange immer wieder, bis jeweils nur noch zwei Elemente übrig sind, die in die richtige Reihenfolge gebracht werden. Hier eine QuickSort-Implementierung in Java:

```
static int[] quickSort(int[] liste) {
   /* quickSort ist nur ein bequemer Starter für die
      eigentliche Arbeits-Methode qSort().          */
   // qSort mit dem gesamten Array aufrufen:
   qSort (liste, 0, liste.length - 1);
```

```
    return liste;
  }

  static void qSort (int[] liste, int lo, int hi) {
    if (lo < hi) {
      // Vergleichselement Median ermitteln:
      median = liste[hi];
      int i = lo - 1;
      int j = hi;
      for ( ; ; ) {
        // Werte von beiden Enden aus vergleichen:
        while (liste[--i] < median);
        while (j > 1 && liste[--j] > median);
        // Schleife verlassen, falls sortiert:
        if (j < i)
          break;
        // Tauschen
        int temp = liste[i];
        liste[i] = liste[j];
        liste[j] = temp;
      }
      liste[hi] = liste[i];
      liste[i] = median;
      // Linke/rechte Partition sortieren:
      qSort (liste, lo, i - 1);
      qSort (liste, i + 1, hi);
    }
  }
```

Die Verwendung von `quickSort()` könnte beispielsweise genauso aussehen wie der weiter oben gezeigte Aufruf von `bubbleSort()`; Sie müssten lediglich den Namen der aufgerufenen Methode ändern und könnten alles andere stehen lassen.

Auffällig sind an dieser Implementierung von QuickSort die etwas gewöhnungsbedürftigen Schleifenkonstruktionen. `for ( ; ; )` ist eine **Endlosschleife**, die allerdings durch den gezielten Aufruf von `break;` verlassen wird. Eine solche Schreibweise ist immer dann nützlich, wenn sich erst während der Ausführung des Schleifenrumpfs herausstellt, dass der weitere Ablauf nicht mehr erforderlich ist.

Ähnlich seltsam sind die beiden in sich geschlossenen `while()`-Schleifen wie diese:

```
while (liste[--i] < median);
```

Da nichts weiter getan werden muss, als die Elemente der Liste nacheinander mit dem Median zu vergleichen, ist dies eine erheblich kompaktere Schreibweise für die folgende Fassung:

```
do {
   --i;
} while (liste[i] < median);
```

6.1.3 Such-Algorithmen

Eine weitere, häufig gestellte Aufgabe bei der Entwicklung von Computerprogrammen ist die Suche nach Elementen in Arrays. Es gibt zwei grundsätzliche Arten dieser Suche: Die lineare Suche durchsucht nacheinander die einzelnen Elemente des Arrays, die binäre Suche erfordert dagegen ein bereits sortiertes Array und sucht darin durch fortgesetztes Halbieren und rekursives Durchsuchen der einzelnen Teilbereiche.

Lineare Suche

Die einfachste Art der Suche ist die lineare. Der Algorithmus besteht einfach darin, die Elemente eines Arrays nacheinander mit einem Suchwert zu vergleichen. Sobald der gesuchte Wert das erste Mal gefunden wird, wird dessen Index zurückgegeben; eine andere Variante würde alle Fundstellen zurückliefern. Es folgt eine Implementierung dieses Algorithmus in Java. Die Methode `suchen()` erwartet die Übergabe eines Arrays von `int`-Werten und den gesuchten Wert. Sie gibt den Index des ersten Vorkommens dieses Werts im Array zurück oder -1, wenn der gesuchte Wert nicht im Array vorkommt. Hier der Code:

```
static int suchen(int[] liste, int wert) {
   for (i = 0; i < liste.length; i++) {
      if (liste[i] == wert)
         return i;
   }
   // Nicht gefunden:
   return -1;
}
```

Dieser Code müsste im Großen und Ganzen selbsterklärend sein. Ein Aufruf dieser Methode könnte folgendermaßen aussehen:

```
public static void main (String args[]) {
   int zahlen[] = {5, 1, 8, 3, 9, 2, 7};
   int suchwert = 8;
   System.out.println (suchwert +
        " kommt in der Liste an Position " +
        suchen (zahlen, suchwert) + " vor.");
}
```

Die Ausgabe dieses Hauptprogramms sieht natürlich so aus:

```
8 kommt in der Liste an Position 2 vor.
```

Binäre Suche

Bei der binären Suche muss zunächst ein sortiertes Array vorliegen. Dieses sortierte Array wird anschließend durch eine rekursive Methode immer weiter unterteilt, dadurch kann der gesuchte Wert sehr schnell gefunden werden.

Hier sehen Sie eine Implementierung dieses Algorithmus in Java. Die Methode `binSuche()` erwartet die Übergabe eines sortierten Arrays (die Sortierung wird nicht überprüft!) und den gesuchten Wert; der Rückgabewert ist wieder der Index des gesuchten Elements oder –1, wenn das Elements nicht gefunden wird:

```
int binSuche (int[] liste, int wert) {
   //Initialisierung der eigentlichen Methode bSuche()
   return bSuche (liste, wert, 0, liste.length);
}

int bSuche (int[] liste, int wert, int lo, int hi) {
   // Nicht gefunden?
   if (lo > hi)
      return -1;
   int m = (lo + hi) / 2;
   int mwert = liste[m];
   if (mwert == wert)
      return wert;
   if (wert < mwert)
      return bSuche (liste, wert, lo, m - 1);
   if (wert > mwert)
      return bSuche (liste, wert, m + 1, hi);
}
```

6.1.4 Ausgewählte Datenstrukturen

Auf der Basis der grundlegenden Datentypen, die im vorigen Kapitel vorgestellt wurden, lassen sich diverse komplexe Datenstrukturen für die Darstellung der verschiedensten Arten von Informationen aus der Realität einrichten. Einige wichtige Datenstrukturen, die häufig für Programmieraufgaben verwendet werden, werden nun vorgestellt.

Stacks und Queues

Die Eigenschaften von **Stacks** (Stapeln) und **Queues** (Warteschlangen) wurden bereits im vorigen Kapitel bei der Vorstellung der Programmiersprache Perl erwähnt. Es handelt sich um Arrays mit einer dynamischen Anzahl von Elementen. Bei einem Stack werden Elemente nacheinander am Ende angehängt und bei Bedarf von diesem Ende wieder entfernt. Es handelt sich also um einen »Last In, First Out« (LIFO)-Speicher. Eine Queue ist dagegen ein »First In, First Out«-Speicher (FIFO): Elemente werden am Anfang der Liste hineingeschoben und am Ende nacheinander entnommen.

In Perl ist die Unterstützung für Stacks und Queues bereits eingebaut. Mit Hilfe der Funktionen `push(@liste, $wert)` und `pop(@liste)` wird ein Stack realisiert: `push` hängt den angegebenen Wert an das Ende der Liste an, `pop` entfernt das letzte Element und gibt es zurück. Mit Hilfe von `unshift` und `pop` lässt sich dagegen eine Queue einrichten: Mit `unshift (@liste, $wert)` wird ein Wert am Anfang einer Liste eingefügt, `pop` entfernt wie gehabt den letzten Wert.

Listen in C

In anderen Programmiersprachen ist die Implementierung dieser Datenstrukturen dagegen ein wenig schwieriger. In C und Java haben die eingebauten Arrays eine feste Anzahl von Elementen, sind also zur Darstellung dynamischer Listen nicht geeignet. In C wird in der Regel ein `struct` definiert, der jeweils einen einzelnen Wert und einen Zeiger auf das nächste dieser Elemente enthält. Eine solche Struktur sieht also beispielsweise so aus:

```
struct element {
   int wert;
   struct element *nachfolger;
}
```

Auch wenn es ein wenig seltsam aussieht, dass innerhalb der `struct`-Definition ein Zeiger auf diesen Datentyp selbst steht, ist dies überhaupt kein Problem. Zwar kann eine Struktur keine Variable enthalten, deren Datentyp die Struktur selbst ist, aber ein Zeiger darauf funktioniert.

Hier ein kleines C-Beispielprogramm, in dem eine Liste verwendet wird, um zehn eingegebene Werte aufzunehmen, die anschließend durch sequenzielles Durchwandern der Liste wieder ausgegeben werden:

```
#include <stdio.h>
#include <stdlib.h>

struct liste {
   int wert;
   struct liste *nachfolger;
};

int main (int argc, char *argv[]) {
   int i;
   int w;
   /* Element erzeugen */
   struct liste *aktuell = malloc(sizeof *aktuell);
   struct liste *erster = aktuell; /* Anfang merken */
   printf ("Bitte 10 int-Werte!\n");
   for (i = 0; i < 10; i++) {
      scanf ("%d", &w);
      aktuell->wert = w;
      aktuell->nachfolger = malloc(sizeof *aktuell);
      aktuell = aktuell->nachfolger;
   }
```

```
    aktuell = erster;
    printf ("\n");
    printf ("Ihre Eingaben:\n");
    for (i = 0; i < 10; i++) {
       printf ("%d\n", aktuell->wert);
       aktuell = aktuell->nachfolger;
    }
    return 0;
}
```

malloc( )

Die Funktion `malloc()` dient dazu, dynamisch Speicher zu allozieren. Das bedeutet, dass ein Bereich im Arbeitsspeicher für einen neuen Wert reserviert wird. Die Speicherverwaltungsfunktionen sind in der Header-Datei `stdlib.h` definiert. `malloc()` besitzt einen etwas merkwürdigen Datentyp: `void *`. Dies ist nicht etwa ein »Zeiger auf gar nichts«, sondern einfach ein Zeiger auf einen beliebigen Datentyp. Damit die richtige Menge Speicher reserviert wird, benötigt `malloc()` die gewünschte Speichermenge (Datentyp `size_t`) als Argument. Hierzu bietet sich der Operator `sizeof` an, der die Speichergröße einer Variablen oder eines Datentyps angibt.

In robusten Programmen für echte Anwendungszwecke sollten Sie `malloc()` stets mittels `if` überprüfen, damit Sie Fälle abfangen können, in denen die Speicherzuteilung nicht möglich ist (etwa wenn insgesamt zu wenig Speicher zur Verfügung steht).

In diesem Beispielprogramm werden zwei verschiedene Zeiger auf ein Listenelement verwendet, damit das erste Element der Liste auch dann noch verfügbar ist, wenn der andere Zeiger zum Durchwandern der Liste genutzt wird.

In Java lässt sich eine Liste noch erheblich einfacher realisieren: Wegen der Objektorientierung können Sie leicht eine Klasse implementieren, die ein Listenelement und einen Verweis auf seinen Nachfolger darstellt. Eine solche Klasse könnte beispielsweise so aussehen:

```
public class Element {
   private int wert;
   private Element nachfolger;
}
```

Es ist leicht vorstellbar, dass sich Funktionen zum Hinzufügen oder Entfernen von Elementen als Methoden einer solchen Klasse implementieren lassen.

Es folgt ein kleines Beispiel für die praktische Anwendung von Listen. Das Programm ist in Perl geschrieben, weil Listen in dieser Sprache bereits eingebaut sind und keinen komplexen Programmieraufwand erfordern. Das Beispiel verwendet einen Stack, um die Wörter in einem String umzukehren:

```
#!/usr/bin/perl -w
use strict;
print "Bitte eine Zeile eingeben:\n";
my $line = <>;
chomp $line;
```

```
# An Leerzeichen zerlegen:
my @words = split (/\s+/, $line);
# Umgekehrte Liste erstellen:
my @rwords;
while (@words) {
   push (@rwords, pop (@words));
}
# Umgekehrte Liste ausgeben:
print join (" ", @rwords);
```

Die Ausgabe des Programms sieht beispielsweise so aus:

```
Bitte eine Zeile eingeben:
Diese Liste wird nun umgekehrt
umgekehrt nun wird Liste Diese
```

die wichtige Anweisung in diesem Programm ist folgende:

```
push (@rwords, pop (@words));
```

Mit Hilfe von `pop` wird jeweils das letzte Zeichen der Liste `@words` entnommen und mittels `push` an die Liste `@rwords` angehängt.

Bäume

Eine weitere interessante Datenstruktur ist der **Baum**. Die einzelnen Elemente eines Baums ähneln denjenigen einer Liste, mit dem Unterschied, dass ein Element eines Baums jeweils mehrere Nachfolger haben kann. Die wichtigste Art der Baumstruktur ist der **Binärbaum**. Jedes Element dieses Baums kann genau zwei Nachfolger besitzen. Mit Hilfe von Binärbäumen werden beispielsweise schnelle Such- und Sortierverfahren implementiert.

Binärbäume in C

In C sieht die Datenstruktur für ein einzelnes Element eines Binärbaums beispielsweise so aus:

```
struct baum {
   int wert;
   struct baum *links;
   struct baum *rechts;
}
```

Hier sehen Sie eine Variante des Listenbeispiels von oben, bei der Eingabewerte zwischen 1 und 100 erwartet werden. Sie werden jeweils nach ihrer Größe in den Binärbaum einsortiert; bei der Ausgabe wird der Baum rekursiv von ganz links nach ganz rechts durchwandert.

```
#include <stdio.h>
#include <mem.h>

struct baum {
   int wert;
```

```
   struct baum *links;
   struct baum *rechts;
};

void zeige_baum (struct baum *aktuell) {
   if (aktuell->links)
      zeige_baum (aktuell->links);
   printf ("%d\n", aktuell->wert);
   if (aktuell->rechts)
      zeige_baum (aktuell->rechts);
}

int main (int argc, char *argv[]) {
   int i;
   int w;
   /* Element erzeugen: */
   struct baum *aktuell = malloc (sizeof *aktuell);
   /* Anfangsposition merken: */
   struct baum *erster = aktuell;
   /* mittleren Wert speichern: */
   aktuell->wert = 50;
   printf ("Bitte 10 Werte von 1 bis 100 eingeben.\n");
   for (i = 0; i < 10; i++) {
      printf ("%2d. ", i + 1);
      scanf ("%d", &w);
      /* Wert einsortieren */
      aktuell = erster;
      for ( ; ; ) {
         if (w <= aktuell->wert) {
            if (aktuell->links) {
               aktuell = aktuell->links;
            } else {
               aktuell->links =
                    malloc (sizeof *aktuell);
               aktuell = aktuell->links;
               break;
            }
         }
         else {
            if (aktuell->rechts) {
               aktuell = aktuell->rechts;
            } else {
               aktuell->rechts =
                    malloc (sizeof *aktuell);
               aktuell = aktuell->rechts;
               break;
```

```
                }
            }
        }
        aktuell->wert = w;
    }
    printf ("\n");
    printf ("Ihre Eingaben:\n");
    zeige_baum (erster);
    return 0;
}
```

Interessant ist, wie dieses Programm die einzelnen Werte in den Baum einsortiert. Für jedes einzelne Element wird bei der Wurzel des Baums begonnen; wenn das Element kleiner oder gleich dem Wert an der Wurzel ist, wird weiter nach links gewandert, andernfalls weiter nach rechts. Auf diese Weise sucht sich jeder Wert seinen passenden Platz im Baum. Je kleiner ein Wert ist, desto weiter links steht er im fertigen Baum.

Die rekursive Funktion `zeige_baum()` wird zunächst mit der Wurzel des Baums aufgerufen. Falls das Baumelement, mit dem sie aufgerufen wurde, einen linken Nachfolger hat, wird sie zuerst für diesen aufgerufen. Anschließend wird der Wert das Element selbst ausgegeben, schließlich wird sie noch einmal für den rechten Nachfolger aufgerufen, falls er existiert. Auf diese Weise wird der Baum nacheinander von links nach rechts ausgegeben.

6.2 Reguläre Ausdrücke

Bereits in Kapitel 4, *Betriebssysteme*, wurden Beispiele für die Mustererkennung in der UNIX-Shell und im Programm `grep` gezeigt. In der Softwareentwicklung spielt die Arbeit mit Textmustern eine sehr wichtige Rolle. Denken Sie nur einmal an die Verarbeitung zahlreicher Personendaten oder Verwaltungsinformationen bei Behörden oder großen Unternehmen. Es wäre mehr als mühselig, solche Datenbestände mit Hilfe konventioneller Einzeltextvergleiche in den Griff bekommen zu wollen.

Deshalb wurde ein besonderes Verfahren entwickelt: die Arbeit mit **regulären Ausdrücken**. Der Begriff »regulärer Ausdruck« ist eigentlich ein wenig unglücklich gewählt. Es handelt sich um die Übersetzung des englischen Wortes »regular expression« (abgekürzt RegExp), das man besser als »Ausdruck mit Regeln« wiedergeben würde – »regulärer Ausdruck« klingt ein wenig wie »gebratener Würstchenverkäufer«.

Reguläre Ausdrücke in Perl

Die beste Unterstützung für reguläre Ausdrücke bietet traditionellerweise die Programmiersprache Perl, denn sie wurde vor allem zu diesem Zweck entworfen. Die Syntax für reguläre Ausdrücke, die in Perl eingesetzt wird, geht weit über die Möglichkeiten von `grep` hinaus. Sie stammen zum großen Teil von den UNIX-Textverarbeitungssprachen `sed` und `awk`. Inzwischen werden reguläre Ausdrücke auch von Java und dem Microsoft-.NET-Framework unterstützt. Da sie das Programmieren erheblich erleichtern können, wird ihnen an dieser Stelle ein ganzer Abschnitt gewidmet.

7.2 Werkzeuge

In diesem Abschnitt werden einige wichtige Hilfsmittel für die Softwareentwicklung vorgestellt:

- Die Diagrammspezifikation **UML** (Unified Modeling Language) bietet verschiedene standardisierte Diagrammtypen zur Darstellung diverser Projektbestandteile in den verschiedenen Phasen des Entwicklungsprozesses.
- **Entwurfsmuster** katalogisieren erfolgreiche Lösungsmodelle für die einfache Übernahme in späteren Projekten.
- **Unit-Tests** ermöglichen den automatisierten Test der Funktionalität einer Klasse.

Diese Hilfsmittel können in vielen unterschiedlichen Entwicklungsmodellen zum Einsatz kommen, obwohl einige von ihnen im Zusammenhang mit konkreten Verfahren entwickelt wurden.

7.2.1 UML

Der Begriff »Sprache« für die **Unified Modeling Language** (UML) ist ein wenig irreführend: Zwar handelt es sich um eine bestimmte Ausdrucksweise, aber ihr Vokabular sind keine Wörter, sondern verschiedene Arten von Diagrammen. Die UML ist gut dazu geeignet, die diversen Aspekte der Softwareentwicklung übersichtlich darzustellen; ihr Arbeitsschwerpunkt sind die Analyse- und die Entwurfsphase.

UML-Geschichte

Die UML wurde in den 90er-Jahren von den »drei Amigos« **Grady Booch**, **Ivar Jacobson** und **James Rumbaugh** durch Zusammenführung einiger früherer Ansätze entwickelt. Zunächst wurde sie vor allem durch den Entwicklungstool-Hersteller **Rational** gefördert und weiterentwickelt. Später wurde sie von der **Object Management Group** (OMG) standardisiert und fand allgemeine Verbreitung. Ende 2003 wurde die neue Version 2.0 veröffentlicht, die einige Erweiterungen und Verbesserungen bietet; sie wurde allerdings noch nicht endgültig verabschiedet, da verschiedene Entwickler und Firmen noch über Teilaspekte diskutieren.

Die wichtigsten **Diagrammtypen** der UML sind die Folgenden:

- **Anwendungsfalldiagramme** (Use Case Diagrams) stellen die Anforderungen der Benutzer dar. Sie prägen das Bild der UML in der Öffentlichkeit, weil sie die typischen »Strichmännchen« (Akteure) definieren.
- **Klassendiagramme** (Class Diagrams) verdeutlichen die Klassenstruktur und damit die Grundarchitektur des Systems.
- In **Kollaborationsdiagrammen** (Collaboration Diagrams) wird die Zusammenwirkung verschiedener Objekte dargestellt.
- **Sequenzdiagramme** (Sequence Diagrams) stellen besonders den zeitlichen Ablauf dieser Zusammenarbeit dar.
- In **Zustandsdiagrammen** (State Diagrams) werden die verschiedenen Zustände eines Objekts und die Übergänge zwischen diesen Zuständen dargestellt.
- **Pakete** (Packages) dienen der Verdeutlichung hierarchischer Objektstrukturen.

- **Aktivitätsdiagramme** (Activity Diagrams) stellen den logischen und zeitlichen Ablauf verschiedener Aktivitäten dar.

UML in der Praxis

Es ist nicht Sinn der Sache, UML-Diagramme auf Dauer mit dem Stift auf Papier zu malen (außer für erste Ideensammlungen oder bei Projektbesprechungen). Sie sollten die Diagramme auch nicht mühevoll in einem allgemeinen Grafikprogramm wie Illustrator zeichnen. Viel verbreiteter und erheblich produktiver sind UML-Tools, mit denen sich die Diagramme einfach, schnell und übersichtlich erstellen lassen. Solche Programme werden auch als **CASE-Tools** (Computer-Aided Software Engineering) bezeichnet. Es gibt sowohl Open-Source-Lösungen als auch sehr teure kommerzielle Produkte.

Ein praktisches Open-Source-Tool ist **ArgoUML** (http://argouml.tigris.org). Es ist in Java geschrieben und läuft somit auf den meisten wichtigen Systemen. Der einzige Nachteil besteht darin, dass das Paket noch nicht die UML-Version 2.0 unterstützt. Eine Installation ist nicht erforderlich; Sie benötigen lediglich eine funktionierende Java-Installation (siehe Kapitel 5, *Grundlagen der Programmierung*). Laden Sie die Archivdatei herunter, entpacken Sie sie in ein beliebiges neues Verzeichnis und geben Sie Folgendes ein, um das Programm zu starten:

```
$ java -jar argouml.jar
```

Arbeit mit ArgoUML

Abbildung 7.2 zeigt das Programm beim Erstellen eines Klassendiagramms. Die obere der beiden Symbolleisten enthält Schaltflächen für die verschiedenen Diagrammtypen, die untere bietet Werkzeuge für die einzelnen Bestandteile des aktuellen Diagramms. Nachdem Sie einen dieser Buttons angeklickt haben, können Sie das Element mit einem weiteren Klick auf der Arbeitsfläche platzieren und im unteren Feld seine Eigenschaften einstellen. Aus dem jeweils ausgewählten Element lassen sich automatisch Verbindungslinien herausziehen, um es mit anderen Elementen zu verknüpfen.

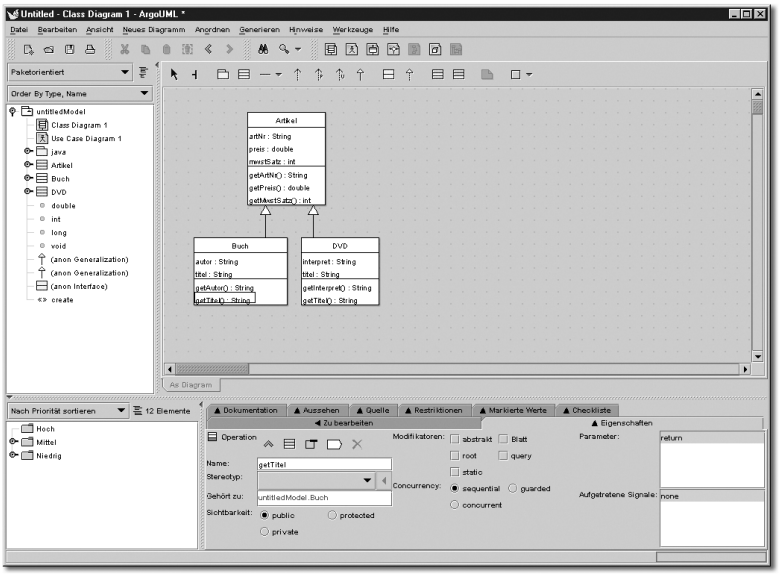

Abbildung 7.2 ArgoUML beim Erstellen eines Klassendiagramms

Im Folgenden werden vier UML-Diagrammarten näher vorgestellt: Anwendungsfalldiagramme, Klassendiagramme, Sequenzdiagramme und Aktivitätsdiagramme.

Anwendungsfalldiagramme

In einem **Anwendungsfalldiagramm** werden Anwendungsfälle (Use Cases) aus der Sicht der Beteiligten, der so genannten **Akteure**, dargestellt. Dieser Diagrammtyp ist vornehmlich ein Bestandteil der Analyse. Abbildung 7.3 zeigt ein Beispiel mit zwei Akteuren, einem Kunden und einem Verkäufer. Die drei möglichen Geschäftsvorfälle, die zwischen ihnen stattfinden können, sind Information, Verhandlung und Kauf.

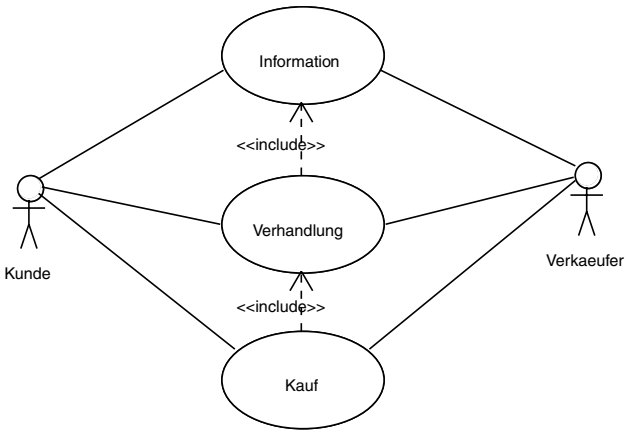

Abbildung 7.3 Ein einfaches UML-Anwendungsfalldiagramm

Die `<<include>>`-Beziehung zwischen Kauf und Verhandlung bedeutet, dass der Verkauf eine Verhandlung umfassen kann; ebenso verhält es sich mit der Beziehung zwischen Verhandlung und Information. Andere mögliche Beziehungen sind:

- Eine durchgehende Linie mit einer hohlen Pfeilspitze steht für eine **Generalisierung**; der Anwendungsfall am Pfeilanfang ist eine Spezialisierung desjenigen am Pfeilende.
- Ein gestrichelter Pfeil mit der Beschriftung `<<extend>>` besagt, dass der Anwendungsfall am Pfeilursprung denjenigen, auf den gezeigt wird, erweitert.

Bei der späteren Implementierung können sich aus diesen Beziehungen Vererbungsverhältnisse ergeben.

Klassendiagramme

Klassendiagramme stellen die Elemente einer Klasse sowie die Beziehungen zwischen verschiedenen Klassen dar. Jede Klasse wird durch ein Rechteck dargestellt; es gibt drei Detailstufen:

- nur Klassenname
- Klassenname und Attribute (Eigenschaften)
- Klassenname, Attribute und Methoden

Die (bis zu) drei Kategorien werden durch waagerechte Linien voneinander getrennt. Abstrakte Klassen (ohne Methodenimplementierung) werden durch `<<abstrakt>>` unter dem Klassennamen gekennzeichnet, Interfaces durch `<<interface>>`.

Vererbung

Die Vererbung wird durch einen Pfeil mit leerer Spitze dargestellt, der von der abgeleiteten auf die übergeordnete Klasse zeigt. In Abbildung 7.4 sehen Sie die Beziehungen zwischen der Oberklasse `Artikel` und den abgeleiteten Klassen `Buch` und `DVD`, jeweils mit allen drei Informationen. Die zusätzliche Angabe der Datentypen von Methoden und Attributen ermöglicht die automatische Erzeugung des Code-Grundgerüsts für die Klassen. In ArgoUML erfolgt dies beispielsweise über **Generieren · Alle Klassen generieren**, sofern gerade ein Klassendiagramm angezeigt wird.

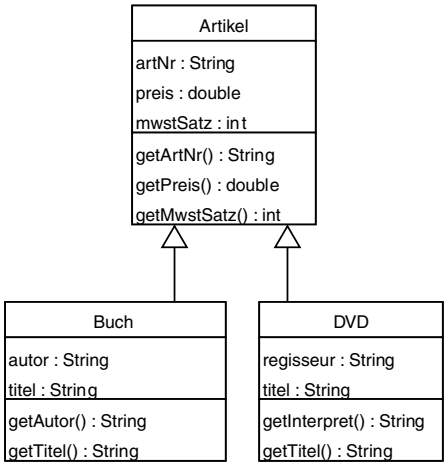

Abbildung 7.4 Ein UML-Klassendiagramm mit einer Elternklasse und zwei abgeleiteten Klassen

Sonstige Beziehungen

Neben den Vererbungslinien gibt es auch komplexere Beziehungen zwischen Klassen beziehungsweise Instanzen. Allgemein spricht man bei der Vererbung von **»IS A«-Beziehungen**, denn eine speziellere Klasse »ist« auch immer die Elternklasse – im vorliegenden Beispiel gilt etwa, dass ein Buch unter anderem ein Artikel ist. Enthält eine Klasse Attribute, die Instanzen einer anderen Klasse sind, spricht man dagegen von einer **»HAS A«-Beziehung** – ein Objekt »hat« oder enthält ein anderes.

In der UML werden »HAS A«-Beziehungen durch eine Verbindungslinie zwischen den beiden Klassen dargestellt; auf der Seite der Klasse, die die andere enthält, wird die Linie durch eine Raute gekennzeichnet. Hier gibt es zwei Möglichkeiten:

- **Aggregation**. Die enthaltenen Elemente existieren unabhängig von der enthaltenden Klasse; dies wird durch eine leere Raute gekennzeichnet.
- **Komposition**. Die enthaltenen Elemente existieren nur in Abhängigkeit von der enthaltenden Klasse, sind also eher abstrakte Eigenschaften als konkrete Gegenstände; diese Beziehung wird durch eine gefüllte Raute dargestellt.

Die **Kardinalität** (Häufigkeit) der beteiligten Elemente kann durch Zahlen an den Verbindungsstellen dargestellt werden:

- Eine einfache Zahl (zum Beispiel 1) bedeutet, dass genauso viele Elemente dieses Typs beteiligt sind. Beispielsweise könnte man die Beziehung zwischen einem PKW und seinen Rädern durch die festen Zahlen 1 beziehungsweise 4 kennzeichnen.
- Ein Sternchen (`*`) steht für beliebig viele, also kein Element, ein Element oder mehrere Elemente.
- Ein Zahlenbereich (`m..n`) bedeutet mindestens `m`, höchstens `n` Elemente. Die Beziehung zwischen einem allgemeinen Fahrzeug und seinen Rädern ist beispielsweise `1` zu `1..*` (von der Schubkarre bis zum beliebig langen Güterzug).

Neben den »HAS A«-Beziehungen gibt es auch diverse lockere Verbindungen zwischen Klassen, genannt **Assoziationen**. Sie werden durch eine Verbindungslinie dargestellt; eine offene Pfeilspitze oder ein daneben gezeichnetes Richtungsdreieck sowie eine Textbezeichnung beschreiben die Beziehung näher. Auch hier können die Kardinalitätsbezeichnungen verwendet werden.

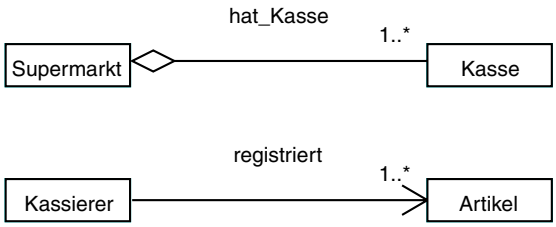

Abbildung 7.5 Klassenbeziehungen in UML-Klassendiagrammen: Aggregation (oben) und Assoziation (unten)

In Abbildung 7.5 finden Sie zwei Beispiele für solche Klassenbeziehungen: Ein Supermarkt kann eine oder mehrere Kassen enthalten; da diese unabhängig vom Supermarkt existieren, bleibt die Raute leer (Aggregation). Die zweite Beziehung zeigt, dass ein Kassierer beliebig viele Artikel registrieren kann. Die Kardinalität 1 bei `Supermarkt` beziehungsweise `Kassierer` wurde als »Standardfall« einfach weggelassen; dies ist eine (zulässige) Eigenart von ArgoUML.

Sequenzdiagramme

In einem **Sequenzdiagramm** wird der zeitliche Ablauf einer Anwendung als Abfolge von Nachrichten zwischen den Objekten dargestellt. Die beteiligten Objekte werden waagerecht auf »Swim Lanes« (Schwimmbahnen) nebeneinander platziert; der Arbeitsablauf wird von oben nach unten dargestellt. Auf jeder Bahn wird eine gestrichelte Linie gezeichnet, solange ein Objekt existiert, oder ein schmales Rechteck, wenn das Objekt gerade den Programmfluss kontrolliert. Methodenaufrufe werden mit Hilfe von durchgezogenen Pfeilen gekennzeichnet, Nachrichten durch gestrichelte Pfeile. Abbildung 7.6 zeigt das einfache Beispiel eines Einkaufsvorgangs.

Die aktuelle Version von ArgoUML (0.16, Anfang Februar 2005) unterstützt leider noch keine Sequenzdiagramme. Verwenden Sie stattdessen die etwas simpleren Kollaborationsdiagramme.

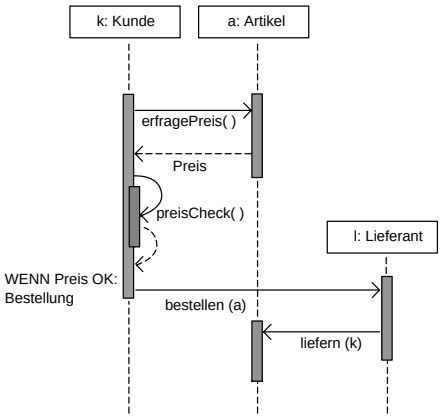

Abbildung 7.6 UML-Sequenzdiagramm eines Einkaufsvorgangs

Aktivitätsdiagramme

Aktivitätsdiagramme lassen sich als Weiterentwicklung der klassischen Flussdiagramme betrachten. Sie dienen der Darstellung des Zusammenspiels verschiedener Aktivitäten, die durch seitlich abgerundete Kästen dargestellt werden. Andere wichtige Symbole sind diese:

- Ein dicker, gefüllter Kreis markiert den Startzustand.
- Ein hohler Kreis, der einen kleineren gefüllten enthält, kennzeichnet den Endzustand, der alle Aktivitäten abschließt.
- Fallentscheidungen werden durch eine Raute mit mehreren abgehenden Pfeilen dargestellt.
- Eine dicke horizontale Linie dient entweder der Aufteilung (Forking) eines Ablaufs in mehrere parallele Verarbeitungsstränge oder der Zusammenführung zuvor verzweigter Abläufe.

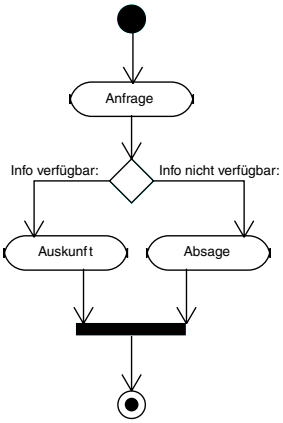

Abbildung 7.7 UML-Aktivitätsdiagramm des Anwendungsfalls »Information«

In Abbildung 7.7 werden mögliche Abläufe des Anwendungsfalls »Information« aus dem Anwendungsfalldiagramm in Abbildung 7.3 verdeutlicht. In ArgoUML können Sie ein Aktivitätsdiagramm eines Anwendungsfalls erstellen, sobald dieser markiert ist.

7.2.2 Entwurfsmuster

Entwurfsmuster (Design Patterns) sind ursprünglich ein Konzept aus der (Gebäude-) Architektur, das im Rahmen der Programmiersprache und -umgebung Smalltalk für die Softwareentwicklung übernommen wurde. Im Wesentlichen geht es um die übersichtliche Katalogisierung einmal gefundener Lösungen für die spätere Wiederverwendung. Beachten Sie, dass Entwurfsmuster keine fertig programmierten Komponenten oder Codeschnipsel sind; wie der Name schon sagt, gehören sie zur Phase des Entwurfs und nicht zur Implementierung von Software. Nichtsdestotrotz enthält ein Muster neben vielen anderen Komponenten auch Codebeispiele.

In der Softwareentwicklung wurden die Entwurfsmuster durch die »Gang of Four« (GoF) **Erich Gamma** (zurzeit Eclipse-Entwicklungsleiter), **Richard Helm**, **Ralph Johnson** und **John Vlissides** eingeführt; ihr Buch »Design Patterns« (siehe Anhang B) ist die wichtigste Informationsquelle und gewissermaßen der Hauptkatalog für alle Entwurfsmuster.

Schema für Entwurfsmuster

Jedes Entwurfsmuster besteht aus vier wesentlichen Komponenten:

- **Name**. Das Muster sollte eine möglichst griffige Bezeichnung erhalten, die möglichst genau auf seinen Verwendungszweck hindeutet.
- **Problem**. Eine genaue Beschreibung der Situation, in der das Entwurfsmuster eingesetzt werden kann
- **Lösung**. Die abstrakte Beschreibung eines Entwurfs, der das Problem löst
- **Konsequenzen**. Eine Beschreibung der Folgen und möglichen Nebeneffekte, die der Einsatz des Patterns mit sich bringt

Muster-Katalogisierung

Pattern-Kataloge wie derjenige im Buch »Design Patterns« verwenden allerdings eine erheblich genauere Struktur zur Beschreibung jedes Musters. Es handelt sich um eine Auflistung der folgenden Punkte (in Klammern jeweils die Originalbezeichnungen aus dem GoF-Buch):

- **Name und Einordnung** (Pattern Name and Classification). Die Bedeutung eines sprechenden Namens braucht unter Programmierern nicht weiter betont zu werden. Die Einordnung beschreibt das Einsatzgebiet (Purpose) und den Geltungsbereich (Scope) des Musters. Man unterscheidet drei grundlegende Einsatzgebiete: **Erzeugungsmuster** (Creational Patterns) sind Lösungen für verschiedene Probleme der Objekterzeugung; **Strukturmuster** (Structural Patterns) beschäftigen sich mit Problemstellungen der Datenstruktur, und **Verhaltensmuster** (Behavioral Patterns) beschreiben die Implementierung häufig benötigter Verhaltensweisen von Objekten. Der Geltungsbereich ist »Klasse« für statische, durch Vererbung angewendete Muster oder »Objekt« für Muster, die Objektbeziehungen betreffen. Letztere kommen wesentlich häufiger vor.

- **Absicht** (Intent). Kurze Beschreibung der Aufgabe des Entwurfsmusters und mögliche Gründe für seinen Einsatz
- **Alias** (Also Known As). Viele Muster sind unter verschiedenen Namen bekannt; andere gängige Bezeichnungen werden hier aufgelistet.
- **Motivation**. Ein konkretes Beispielszenario, das den Einsatzzweck des Musters deutlich macht
- **Verwendungszweck** (Applicability). Beschreibung der Situationen, in denen das Pattern eingesetzt werden kann, und der Probleme, die es lösen hilft
- **Struktur** (Structure). Grafische Darstellung der Klassen des Entwurfsmusters, meist UML-basiert
- **Beteiligte** (Participants). Klassen und Objekte, die in die Anwendung des Musters involviert sind
- **Zusammenspiel** (Collaborations). Beschreibung der Zusammenarbeit zwischen den Beteiligten
- **Konsequenzen** (Consequences). Ergebnisse sowie Vor- und Nachteile der Anwendung des Musters
- **Implementierung** (Implementation). Beschreibung von Besonderheiten und möglichen Problemen bei einer Implementierung des Musters
- **Codebeispiele** (Sample Code). Das GoF-Buch verwendet C++ und/oder Smalltalk; in neueren Büchern und Websites wird meist Java oder C# benutzt. Prinzipiell kann jede objektorientierte Programmiersprache zum Einsatz kommen.
- **Einsatzbeispiele** (Known Uses). Beispiele für die Anwendung dieser Muster in realen Softwaresystemen
- **Querverweise** (Related Patterns). Zusammenarbeit dieses Entwurfsmusters mit anderen Mustern; gegebenenfalls Gemeinsamkeiten und Unterschiede

Das Buch »Design Patterns« enthält einen Katalog von dreiundzwanzig Mustern, die nach diesem Schema aufgelistet werden.[6] Es handelt sich um Probleme, vor denen eines Tages jeder steht, der größere objektorientierte Programme schreiben möchte. Es hält Sie allerdings nichts davon ab, Ihre eigenen gelungenen Lösungsansätze ebenfalls nach diesem Schema zu katalogisieren. Wenn Sie einem Entwicklungsteam angehören, könnten Ihre Kollegen Ihnen eines Tages dafür dankbar sein.

Ein konkretes Beispiel

Nach dem oben beschriebenen Schema wird hier ein konkretes Beispiel für ein Entwurfsmuster gezeigt: das **Singleton-Pattern**. Es geht um eine Klasse, von der nur genau eine einzige Instanz existieren darf.

- **Name**: Singleton
- **Einordnung**: Erzeugungsmuster, Objekt

6 Eine vollständige Liste, größtenteils mit Verweisen auf kurze Beschreibungen, enthält der Artikel »Entwurfsmuster« in der Wikipedia (**http://de.wikipedia.org/wiki/ Entwurfsmuster**).

- **Absicht**: Sicherstellen, dass eine Klasse nur genau eine einzige Instanz hat
- **Alias**: keins (deutsche Bezeichnung: Einzelstück)
- **Motivation**: Bestimmte Objekte darf es selbst im größten System nur ein einziges Mal geben. Denken Sie beispielsweise an eine zentrale Warteschlange für Datei-, Drucker- oder Netzwerkzugriffe. Praktischerweise wird ein solches Element als Klasse erstellt, die nur beim ersten Aufruf eine neue Instanz erzeugt und bei späteren Aufrufen immer wieder einen Verweis auf diese Instanz zurückgibt. So brauchen Sie beim Aufruf nicht mehr zu überprüfen, ob die Instanz bereits existiert.

  Daneben kann der Singleton auch als bessere globale Variable dienen, weil die Instanz auf einfache Weise global verfügbar ist.
- **Verwendungszweck**: Benutzen Sie dieses Muster, wenn Sie eine erweiterbare Klasse brauchen, die ohne Modifikation des aufrufenden Codes nur genau eine Instanz besitzen darf.
- **Struktur**: Die Struktur der Klasse `Singleton` wird in Abbildung 7.8 dargestellt.

Abbildung 7.8 UML-Struktur der Klasse Singleton und ihrer einzigen Instanz

- **Beteiligte**: Das einzige Element ist die Klasse `Singleton` selbst, die auf Anforderung ihre einzige Instanz zurückgibt und gegebenenfalls neu erzeugt.
- **Zusammenspiel**: Andere Klassen rufen die Methode `instance()` auf, um die einzige Instanz der Klasse zu erhalten.
- **Konsequenzen**: Das Entwurfsmuster Singleton bietet eine Reihe von Vorteilen gegenüber anderen Lösungen. Hier die wichtigsten Beispiele:
  - Die Verwendung einer globalen Variablen wird vermieden; dies beseitigt eine potentielle Fehlerquelle.
  - Statt genau einer Instanz können Sie mit Hilfe dieses Musters auch eine beliebige andere (festgelegte) Anzahl von Instanzen zulassen.
  - Die Klasse bleibt erweiterbar – im Gegensatz zu anderen Lösungsansätzen für dieses Problem können problemlos abgeleitete Klassen gebildet werden.
- **Implementierung**: Die Instanz wird zum statischen Attribut der Klasse selbst und mit `null` (noch keine Instanz vorhanden) initialisiert. Der Konstruktor wird mit der Veröffentlichungsstufe `private` versehen, so dass er nicht von außen aufgerufen werden kann. Die öffentliche Methode `instance()`, die Clients stattdessen aufrufen können, überprüft zunächst, ob die Instanz bereits erzeugt wurde; falls nicht, ruft sie den Konstruktor auf. Anschließend wird in jedem Fall eine Referenz auf die Instanz zurückgegeben.

- **Codebeispiele**: Die Implementierung der Klasse `Singleton` ist nicht besonders umfangreich. Hier eine vollständige Java-Klasse, die dem Entwurfsmuster genügt:

```
public class Singleton {

   // Die Instanz - zunächst noch nicht vorhanden
   private static Singleton singleton = null;

   // der private Konstruktor
   private Singleton() {
   }

   // die Client-Methode instance()
   public static synchronized Singleton instance() {
      // Instanz erzeugen, falls noch keine existiert
      if (singleton == null) {
         singleton = new Singleton();
      }
      // Instanz auf jeden Fall zurückgeben
      return singleton;
   }
}
```

Der Modifikator `synchronized` bei der Methode `instance()` ist wichtig: Wenn mehrere Threads die Methode parallel aufrufen, könnten sonst versehentlich doch mehrere Instanzen erzeugt werden.

- **Einsatzbeispiele**: Unzählige – alle künstlichen »Engpässe« wie beispielsweise Warteschlangen folgen diesem Schema.
- **Querverweise**: Entwurfsmuster wie Abstract Factory (eine Art Multikonstruktor für mehrere abgeleitete Klassen), Builder und Prototype lassen sich mit Hilfe des Singleton-Patterns implementieren.

7.2.3 Unit-Tests

Wenn die Zeit in Softwareentwicklungsprozessen eng wird, verzichten die Entwickler am ehesten auf ausgiebige Tests. Das ist fatal für die Qualität der veröffentlichten Anwendungen – der Extremfall ist so genannte Banana Ware, die »grün« ausgeliefert wird und erst »beim Kunden reift«. Teure kommerzielle Software ist sogar häufiger von diesem Problem betroffen als Open-Source-Projekte, weil die Marketing- und Vertriebsabteilungen oftmals massiven Druck auf die Entwicklungsteams ausüben, um angekündigte Veröffentlichungstermine einzuhalten.

Motivation

Natürlich haben Entwickler sich seit Jahrzehnten Gedanken darüber gemacht, wie sich die unbefriedigende Situation im Bereich der Softwaretests verbessern ließe. Eine wichtige Erkenntnis ist, dass Programmcode sich am exaktesten durch weiteren Programmcode überprüfen lässt. Anstatt sich also den Kopf darüber zu zerbrechen, welche Fehler auftre-

ten könnten, und mühsam von Hand entsprechende Zustände herbeizuführen, sollten Sie einen automatisierten Test schreiben und mit verschiedenen Werten durchlaufen lassen.

Die neueste Lösung zur Testautomatisierung sind die so genannten **Unit-Tests** oder auch Klassentests. Für beinahe jede ernstzunehmende Programmiersprache steht inzwischen ein xUnit-Framework zur Verfügung, das die Durchführung von Tests vereinfacht und beschleunigt und diese so zu einem integralen Bestandteil der Programmierarbeit macht. Der Klassiker ist das hier vorgestellte **JUnit**-Framework für Java, das von Erich Gamma und Kent Beck geschrieben wurde.

JUnit einrichten

Der erste Schritt besteht darin, JUnit herunterzuladen und in Betrieb zu nehmen. Besuchen Sie die Projekt-Website **http://www.junit.org** und laden Sie sich das aktuelle Paket herunter (zurzeit `junit3.8.1.zip`). Entpacken Sie das Archiv in ein beliebiges Verzeichnis und erweitern Sie Ihren `CLASSPATH` so, dass er `junit.jar` aus diesem Verzeichnis enthält. Nun können Sie das Framework sofort einsetzen.

Ein einfaches Testbeispiel

Betrachten Sie als Beispiel die folgende Klasse, `Artikel`. Sie enthält drei Methoden, die den Bruttopreis, den Mehrwertsteuerbetrag und den Nettopreis zurückgeben sollen:

```
public class Artikel {

   private double preis;
   private int mwst;

   public Artikel () {
      this.preis = 0;
      this.mwst = 16;
   }

   public Artikel (double p, int m) {
      this.preis = p;
      this.mwst = m;
   }

   public double getBrutto () {
      return this.preis;
   }

   public double getMwst () {
      return this.preis / (100 + this.mwst) * this.mwst;
   }

   public double getNetto () {
      return this.preis - this.getMwst();
   }
}
```

Ein Test Angenommen, Sie möchten die ordnungsgemäße Funktion der Methode `getNetto()` testen. Dazu können Sie mit Hilfe von JUnit folgenden Unit-Test schreiben:

```
import junit.framework.*;

public class ArtikelTest extends TestCase {

   public ArtikelTest (String name) {
      super (name);
   }

   public void testNetto() {
      Artikel a1 = new Artikel (116, 16);
      assertTrue (a1.getNetto() == 100);
   }

   public static void main(String[] args) {
      junit.swingui.TestRunner.run (ArtikelTest.class);
   }
}
```

Das Package `junit.framework` enthält die JUnit-Testklassen, die hier verwendet werden. Jeder Test sollte die Klasse `TestCase` erweitern. Wichtig ist hier ein Konstruktor, der einen String als Parameter erwartet; Sie können ihn, wie im Beispiel, einfach an den entsprechenden Konstruktor der übergeordneten Klasse weiterreichen.

Testausführung `main()` ruft die Methode `run()` von `junit.swingui.TestRunner` auf, sodass beim Ausführen der Klasse automatisch die GUI-Variante von JUnit ausgeführt wird. `run()` sorgt automatisch für die Ausführung sämtlicher Methoden, die mit `test*()` beginnen. Dabei gibt es ein sehr deutliches Zeichen für Erfolg oder Misserfolg:

- Ein **grüner Balken** zeigt, dass der Test bestanden wurde.
- Ein **roter Balken** bedeutet Misserfolg; im unteren Fensterabschnitt werden die entsprechenden Fehlermeldungen angezeigt.

Der Test selbst funktioniert folgendermaßen: Die Annahme ist, dass ein Bruttopreis von 116,– € beim üblichen Mehrwertsteuersatz von 16 % zu einem Nettopreis von 100,– € führt. Also wird die Testmethode `assertTrue()` mit dem Vergleich zwischen dem Ergebnis von `getNetto()` und dem Wert 100 aufgerufen. Falls die Vermutung richtig sein sollte, führt sie zu einem grünen Balken – in Abbildung 7.9 wird dieses erfreuliche Ergebnis gezeigt. Das Framework definiert übrigens noch weitere `assert*()`-Methoden, weil Erfolg nicht in jedem Fall durch ein »richtiges« (oder besser wahres) Ergebnis angezeigt wird.

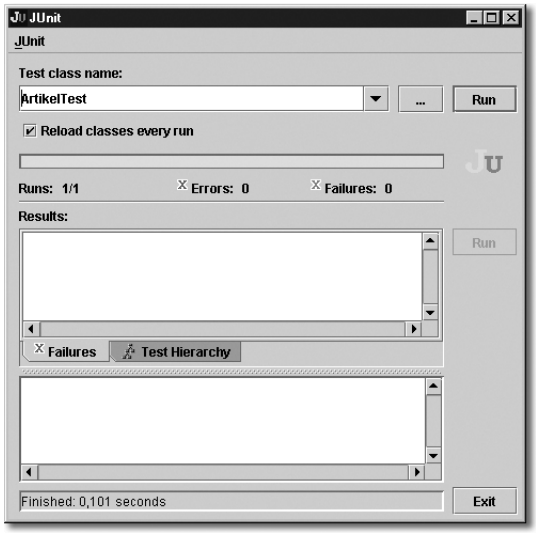

Abbildung 7.9 Erfolgreicher Test der Klasse Artikel in JUnit

Das »Test first«-Verfahren

Damit der Test auf keinen Fall »vergessen« werden kann, empfiehlt es sich, ihn nicht etwa nach der Implementierung eines Features, sondern vorher zu schreiben. Das hat natürlich zur Folge, dass er zunächst nicht bestanden wird. Daraus ergibt sich die Arbeitsweise des **Test-driven Development** (TDD), zu Deutsch testgetriebene Entwicklung:

- **Red** – einen Test schreiben, der zunächst fehlschlägt (roter Balken)
- **Green** – Code schreiben, der den Test mit den einfachsten möglichen Mitteln besteht
- **Refactor** – den neuen Code durch Refactoring vernünftig integrieren, zum Beispiel unnötige Doppelanweisungsfolgen vermeiden

Test-first-Beispiel

Statt durch viel Theorie lässt sich der Test-first-Ansatz am einfachsten an einem Beispiel zeigen: Die Klasse `Artikel` aus dem vorigen Unterabschnitt soll um eine Methode namens `getDMBrutto()` erweitert werden, der den Bruttopreis in DM ausgibt (praktisch für die besonders in der Vorweihnachtszeit häufig anzutreffenden »Zahl mit DM«-Aktionen).

Zunächst wird also ein einfacher Test geschrieben:

```
import junit.framework.*;

public class DMTest extends TestCase {

   public DMTest (String name) {
      super (name);
   }
```

```
    public void testDMBrutto() {
       Artikel a = new Artikel (100, 16);
       assertTrue (a.getDMBrutto() == 195.583);
    }

    public static void main(String[] args) {
       junit.swingui.TestRunner.run (DMTest.class);
    }
}
```

Da der Umrechnungsfaktor bekannt ist, lässt sich leicht vorhersagen, welcher Wert für 100,– € herauskommen muss. Diese Vermutung wird als Testfall formuliert.

Damit der Test sich überhaupt kompilieren lässt, wird zumindest ein »Dummy« der `Artikel`-Methode `getDMBrutto()` benötigt. Dieser könnte beispielsweise so aussehen:

```
public double getDMBrutto () {
   return 0;
}
```

Red In einem so offensichtlichen Fall bräuchten Sie den Test noch nicht einmal auszuführen, um zu wissen, dass er scheitern wird. Tun Sie es trotzdem – nur so gewöhnen Sie sich an den »Test first«-Ablauf.

Green Der nächste Schritt besteht darin, sicherzustellen, dass der Test bestanden wird. Der erste Ansatz darf ruhig eine »Brute Force«-Methode (rohe Gewalt) sein. Wenn beispielsweise `195.583` verlangt wird, kann dieser Wert einfach zurückgegeben werden:

```
public double getDMBrutto () {
   return 195.583;
}
```

Refactor Nun erscheint der erwartete grüne Balken. Jetzt braucht die Methode nur noch verallgemeinert zu werden, damit er für beliebige Werte das richtige Ergebnis liefert; das ist die für diesen Fall geeignete Form des Refactorings. So ergibt sich folgende Endfassung:

```
public double getDMBrutto () {
   return this.preis * 1.95583;
}
```

Nach dem Refactoring sollten Sie den Test natürlich noch einmal durchführen, um sicherzustellen, dass Sie sich nicht vertan haben.

Auf diese Weise können Sie ein Projekt Test für Test aufbauen; in seinem Buch »Test driven Development by Example« vergleicht Kent Beck diese Arbeitsweise mit dem Heraufziehen eines Eimers aus einem Brunnen, bei dem die Kurbelwelle mit Zähnen ausgestattet ist, die beim Loslassen einrasten. Genau dies verspricht das Test-driven Development: Zu jeder Zeit »Clean Code that works«, also jederzeit ein so gut wie releasefähiges Projekt.

Einen guten Einstieg in die Arbeit mit JUnit bietet der lesenswerte Aufsatz »Test-Infected: Programmers love writing tests« (**http://junit.sourceforge.net/doc/testinfected/testing.htm**), der übrigens auch mit der Offline-Dokumentation von JUnit geliefert wird.

9 Bildbearbeitung und Grafik

Leben allein genügt nicht, sagte der Schmetterling. Sonnenschein, Freiheit und eine kleine Blume muss man haben.
– Hans Christian Andersen

Das Bearbeiten von Bildern und Grafiken ist ein wesentlicher Bestandteil der Medienproduktion. Druckerzeugnisse, Multimedia-Projekte, Webseiten und viele Arten von Software werden mit Abbildungen, grafischen Darstellungen und Schmuckelementen ausgestattet. Dieses Kapitel beschäftigt sich mit der Bearbeitung der beiden Hauptarten von Computergrafik:

- Die **Pixelgrafik**, auch **Bitmap-Grafik** genannt, speichert Bilder in Form einzelner farbiger Bildpunkte oder Pixel. Bitmaps besitzen eine bestimmte Auflösung, die ihre Darstellungsqualität und ihren Detailreichtum bestimmen. Pixelgrafik wird vor allem für sämtliche Arten digitalisierter Bilder verwendet, also für Scans und Fotos aus Digitalkameras.
- Bei der **Vektorgrafik** werden die Kurven und Linien durch mathematische Formeln beschrieben und abgespeichert. Das führt dazu, dass Vektorgrafik auflösungsunabhängig ist, weil sich die Kurven und Linien auf jede beliebige Anzahl von Pixeln umrechnen lässt. Diese Art der Grafik ist gut für Zeichnungen oder Diagramme geeignet.

Überblick über Grafiksoftware

Für die Bearbeitung beider Arten der Grafik gibt es sehr professionelle Software. In diesem Kapitel wird für Bitmap-Grafik das Programm **Adobe Photoshop** beschrieben, für Vektorgrafik **Adobe Illustrator**.

Selbstverständlich gibt es eine Reihe von Alternativen auf dem Markt. Pixelgrafik lässt sich beispielsweise auch mit PaintShop Pro oder Ulead PhotoImpact bearbeiten. Speziell für das Web ist das Programm Macromedia Fireworks geeignet; es wird in Kapitel 16, *HTML, CSS und Webdesign*, vorgestellt[1]. Interessant ist außerdem das aus der Linux-Welt stammende freie Bildbearbeitungsprogramm GIMP – sein Funktionsumfang entspricht beinahe dem von Photoshop. Inzwischen ist das Programm auch für Windows verfügbar, unter **www.gimp.org** können Sie es mitsamt Dokumentation herunterladen.

Außerdem bietet Adobe selbst eine abgespeckte Version von Photoshop an, genannt Photoshop Elements – sie bietet wichtige Funktionen von Photoshop zum Preis von etwa 80 € (gegenüber ungefähr 1.000 € für den »großen« Photoshop), ist aber für professionelle Printproduktionen nicht geeignet.

Für Vektorgrafik gibt es als Alternativlösungen beispielsweise Macromedia FreeHand (vergleichbarer Funktionsumfang) oder das leider ziemlich unübersichtliche Corel Draw.

1 Das ausschließlich als Paketbestandteil von Photoshop gelieferte Adobe ImageReady ist Fireworks recht ähnlich.

9.1 Theoretische Grundlagen

In diesem kurzen Abschnitt werden einige wichtige Grundbegriffe der Grafik erläutert, deren Kenntnis für das Verständnis von Bildbearbeitungs- und Zeichenprogrammen sinnvoll ist. Es geht vor allem um Farben und um die Bildauflösung.

9.1.1 Licht und Farbe

Damit Sie die Funktionsweise von Computergrafik verstehen können, folgt hier als Erstes eine kurze Einführung in die Phänomene Licht und Farbe, auf denen die grafischen Darstellungen basieren, die Sie auf dem Bildschirm und später im Ausdruck sehen.

Lichtwellen

Elektromagnetische Wellen eines bestimmten Wellenlängenbereichs können vom menschlichen Auge wahrgenommen werden; diese Wellen werden als **Licht** bezeichnet. Je nach Wellenlänge werden diese Lichtwellen als verschiedene Farben gesehen. Die längsten sichtbaren Lichtwellen haben eine Länge von etwa 800 nm (Nanometer; 1 nm = 10^{-6} m) und werden als rot wahrgenommen, die kürzesten sind ungefähr 400 nm lang und erscheinen violett. Dazwischen liegt das so genannte Farbspektrum mit der vom Regenbogen bekannten Farbfolge rot, orange, gelb, grün, cyan, blau und violett. Wellen, die länger sind als rotes Licht, heißen infrarot, sind sie kürzer als violett, heißen sie ultraviolett. Sehr kurze Wellen sind beispielsweise Röntgen- oder Gammastrahlen; verhältnismäßig lang sind dagegen etwa Radiowellen.

Weißes Licht, wie es etwa die Sonne abstrahlt, ist die gleichmäßige Mischung des gesamten Farbspektrums. Durch ein Prisma lässt sich dieses Licht in die einzelnen Farben des Spektrums zerlegen. Genau dieser Effekt ist für den Regenbogen verantwortlich, denn die Regentropfen dienen dabei als Prisma.

Lichtstrahlen sind übrigens das Einzige, was Sie wirklich sehen können. Wenn Sie einen Gegenstand, eine Person oder ein Bild betrachten, sehen Sie eigentlich nur das Licht, das von ihnen reflektiert wird. Die verschiedenen Farben von Objekten kommen dadurch zustande, dass verschiedene Oberflächen nur einzelne Teile des Farbspektrums reflektieren, andere aber absorbieren.

Wenn Sie also zum Beispiel eine Pflanze als grün wahrnehmen, liegt das daran, dass sie nur grünes Licht reflektiert und alle anderen Farben des Spektrums absorbiert. Genau dasselbe gilt für ein Bild, das auf Papier gedruckt wurde. Betrachten Sie dagegen ein Bild auf einem Monitor oder Fernseher, dann nehmen Sie direkt die Lichtstrahlen wahr, die das Gerät aussendet.

Color-Management

Diese unterschiedliche Art der Erzeugung von Farben auf dem Monitor und im Druck sorgt bei der Bearbeitung von Grafiken und Bildern für Printproduktionen für gewisse Schwierigkeiten: Es ist so gut wie unmöglich, zu erreichen, dass die Farben auf dem Monitor genauso aussehen wie später im Druck. Es ist ein großes Maß an Erfahrung notwendig, um dennoch die richtigen Farbtöne auszuwählen. Moderne Hard- und Software versucht zwar, das Aussehen der Farben über das so genannte Color-Management geräteunabhängig identisch zu halten, aber letztlich hängt das Ganze von zu vielen verschiedenen Faktoren ab. Eine wichtige Rolle spielt beispielsweise die Beleuchtung im Arbeitsraum. Auch

Bauart und Fabrikat des verwendeten Monitors sind entscheidend; darüber hinaus verändern sich die Farbeigenschaften von Monitoren mit zunehmendem Alter.

Farbmodelle

Die Art und Weise, wie Farben im Computer gespeichert werden, hängt vom jeweiligen Verwendungszweck ab. Konkret orientieren sich die so genannten Farbmodelle, auf denen die Speicherung der Farben basiert, am Aufbau des Monitorbildes beziehungsweise an der traditionellen Zusammensetzung von Farben für den Druck.

Farbaddition

Konkret bestehen Monitorbilder aus kleinen Punkten in drei verschiedenen Farben: Rot, Grün und Blau. Alle drei Einzelpunkte bilden zusammen einen **Pixel**, also einen einzelnen Bildpunkt, der mit einer beliebigen Farbe eingefärbt werden kann. Jede der drei Farbkomponenten kann mit unterschiedlicher Leuchtkraft strahlen. Wenn alle drei ihre volle Leuchtkraft erreichen, ergeben sie zusammen weißes Licht. Sind sie dagegen alle ausgeschaltet, sehen Sie an dieser Stelle die Farbe Schwarz. Dieses Verfahren der Mischung von Lichtfarben wird **Farbaddition** genannt, weil von Schwarz aus immer mehr Farbe hinzuaddiert wird, bis schließlich Weiß entsteht.

Beim Druck geschieht im Grunde genommen das Gegenteil. Die so genannten Körper- oder Pigmentfarben, die zum Drucken verwendet werden, werden auf weißes Papier aufgetragen. Als Druckfarben werden die drei Komplementärfarben (gegenteilige Farben) der Lichtfarben verwendet: Cyan (die Komplementärfarbe zu Rot), Magenta (zu Grün) und Gelb (zu Blau).

Rein rechnerisch müsste das Übereinanderdrucken aller drei Druckgrundfarben mit voller Intensität Schwarz ergeben. Dass dies nicht der Fall ist, liegt an den physikalischen Eigenschaften der verwendeten Farbstoffe: Sie weichen stets ein wenig von mathematisch korrektem Cyan, Magenta und Gelb ab. Das Übereinanderdrucken ergibt daher kein Schwarz, sondern irgendein dunkles Grau-Braun. Den Effekt können Sie bei einigen besonders billigen älteren Tintenstrahldruckern beobachten, die nur mit den drei Farben Cyan, Magenta und Gelb drucken.

So gut wie alle höherwertigen Drucker und Druckmaschinen benutzen deshalb in der Praxis echtes Schwarz als vierte Druckfarbe, um Text, Konturen und Kontraste klarer darstellen zu können. Auf diese Weise entsteht der bekannte **Vierfarbdruck** (4C).

Farbsubtraktion

Die Mischung der Druckfarben auf Papier wird als **Farbsubtraktion** bezeichnet, weil die Helligkeit durch das Auftragen von Farbe immer weiter nachlässt.

Für die Speicherung von Farbbildern im Computer entstehen aus diesen Vorgaben zwei verschiedene Farbmodelle: RGB und CMYK.

Für die Bildschirmdarstellung werden Farben im **RGB**-Modus gespeichert, also als Mischung von Rot, Grün und Blau mit unterschiedlicher Intensität. Wird für alle drei Farben der Wert 0 gespeichert, ergibt sich Schwarz; wenn dagegen alle mit ihrem maximalen Wert gespeichert werden, entsteht Weiß. Wie hoch dieser maximale Wert konkret ist, hängt von der verwendeten Farbtiefe ab.

Allgemein gilt, dass derselbe Anteil aller drei Farben Grau ergibt. Je stärker die Werte für die drei Komponenten voneinander abweichen, desto höher ist die **Sättigung** der resultierenden Farbe – sie erscheint »bunter«.

Die Farbcodierung für den Vierfarbdruck ist der **CMYK**-Modus, die Mischung aus den vier Komponenten Cyan, Magenta, Gelb (Yellow) und Schwarz (Black; da B für Blau steht, wird hier der letzte Buchstabe verwendet). Das Farbmodell ist ein wenig komplexer als das RGB-Modell, da es durch die zusätzliche Verwendung von Schwarz häufig mehrere Möglichkeiten gibt, eine bestimmte Farbe darzustellen. Wenn bei diesem Farbmodell alle vier Farben den Wert 0 aufweisen, entsteht Weiß. Wenn Cyan, Magenta und Gelb mit ihrem Maximalwert gespeichert werden, ergibt dies rechnerisch Schwarz; der höchste Wert im Schwarzkanal bedeutet natürlich ebenfalls Schwarz.

Farbseparation

Die Abweichung der Mischung aus Cyan, Magenta und Gelb von reinem Schwarz wird in guten Bildbearbeitungsprogrammen zusammen mit anderen Farbeigenschaften in einem so genannten Farbprofil für das jeweilige Ausgabegerät gespeichert. Dieses Profil dient als Grundlage für die Umrechnung der Monitor-RGB-Darstellung in CMYK. Dieser Umrechnungsprozess wird als Farbseparation bezeichnet.

Es existieren zwei verschiedene Grundarten der Farbseparation: Bei **UCR** (Under Color Removal) werden Cyan, Magenta und Gelb in einem Maß herausgenommen, in dem sie identisch sind, und durch einen entsprechenden Schwarzanteil ersetzt. Dies betont die Tiefen der Farben besser, und es wird insgesamt weniger Druckfarbe benötigt. Bei **GCR** (Grey Component Replacement) werden dagegen auch bestimmte farbige Anteile der drei anderen Farben durch Schwarz ersetzt, was dunklere und stark gesättigte Farben verbessert.

Beachten Sie, dass es neben dem Vierfarbdruck auch noch ein Druckverfahren mit separat hergestellten Pigmentfarben gibt, den so genannten **Volltonfarben**, die auch **Schmuck-** oder **Sonderfarben** heißen. Der unmittelbare Druck solcher Farben ist mitunter erforderlich, da sich nicht jede Farbe des sichtbaren Spektrums durch CMYK-Farben wiedergeben lässt. Sonderfarben gibt es von diversen Herstellern, bekannte Sonderfarbskalen sind etwa HKS oder Pantone. Besonders wichtig sind Sonderfarben, um beispielsweise Metalltöne oder die speziellen Hausfarben mancher Unternehmen drucken zu können. Sämtliche Farben, die sich mit Hilfe von CMYK darstellen lassen, werden dagegen als **Prozessfarben** bezeichnet.

Weitere Farbmodelle

Neben RGB und CMYK, die die Besonderheiten des Monitorbildes beziehungsweise des Ausdrucks berücksichtigen, sind noch zwei weitere Farbmodelle gebräuchlich:

- Beim **HSB**-Modell (Hue, Saturation and Brightness) wird eine Farbe durch die drei Informationen Farbton (Hue), Farbsättigung (Saturation) und Helligkeit (Brightness) dargestellt. Der Farbton wird als Winkel in einem Kreis angegeben, in dem das gesamte Farbspektrum abgebildet ist. Die Sättigung entscheidet, wie blass oder kräftig eine Farbe erscheint; bei einem Sättigungsgrad von 0 ist sie grau. Die Helligkeit schließlich wird in Prozent angegeben: 0 % ist Schwarz und 100 % Weiß.
- Das **Lab**-Modell ist das umfangreichste Farbmodell, RGB und CMYK sind Teilmengen davon. Bei Lab werden zwei Achsen durch den Farbkreis gezogen. Die Achse a verläuft von Rot nach Grün, während b von Blau nach Gelb reicht. Die Farbe wird auf dem Farb-

kreis durch den Schnittpunkt dieser Koordinaten dargestellt. Die Helligkeit (L), der dritte Wert, funktioniert wie gehabt. Der größte Vorteil von Lab ist die Geräteunabhängigkeit, jede Farbe wird exakt und eindeutig beschrieben.

Die Vorstufe des Lab-Modells wurde bereits 1931 von der Commission Internationale d'Eclairage (CIE), einem internationalen Gremium, standardisiert. Es entspricht dem Farbempfinden der meisten Menschen und wird daher als Referenzmodell für Farbkorrekturen eingesetzt. Abgesehen davon rechnen hochwertige Bildbearbeitungsprogramme wie Photoshop intern mit Lab-Farben, damit die Bilder jederzeit verlustfrei in RGB oder CMYK konvertiert werden können.

Die Farbtiefe

Ein weiteres Merkmal, das die für ein digitales Bild gespeicherten Farben betrifft, ist die **Farbtiefe**. Sie besagt, welche Datenmenge für einen einzelnen Pixel gespeichert wird, bestimmt also die Anzahl unterschiedlicher Farben im Bild.

Für RGB-Bilder wird üblicherweise eine Farbtiefe von 24 Bit verwendet: Pro Grundfarbe werden 8 Bit gespeichert, es gibt also je 256 verschiedene Abstufungen von Rot, Grün und Blau. Damit lassen sich über 16,7 Millionen unterschiedliche Farben darstellen. Wenn 32 Bit zur Verfügung stehen, kann zusätzlich zu den drei Farbwerten auch noch ein **Alphakanal** verwendet werden, der den Transparenzgrad für jeden einzelnen Pixel angibt. Bei CMYK-Bildern werden 32 Bit für das eigentliche Bild verwendet, weil jede der vier Grundfarben 8 Bit benötigt. Wirklich genutzt wird diese Datentiefe in der Praxis allerdings nicht, weil die CMYK-Anteile üblicherweise in Prozent und nicht als 8-Bit-Werte zwischen 0 und 255 angegeben werden.

Einige Bilddateiformate beschränken übrigens die Farbtiefe für das gesamte Bild auf deutlich geringere Bit-Breiten. Es gibt beispielsweise Bilder mit nur 16 Bit Farbtiefe (insgesamt 65.536 Farben), wodurch weniger verschiedene Mischungen der jeweiligen Grundfarben möglich sind.

Ein etwas anderer Weg wird bei Bildern beschritten, die insgesamt nur eine Farbtiefe von 8 Bit besitzen: Sie sind jeweils mit einer eigenen Farbpalette oder -tabelle ausgestattet, die die RGB-Werte der verwendeten Farben definiert. Ein solches Bild enthält also maximal 256 Farben, aber sie können beliebig gewählt werden. Farben aus einer durchnummerierten Palette werden als **indizierte Farben** bezeichnet.

Ob die verwendete Farbtiefe überhaupt auf dem Bildschirm angezeigt werden kann, hängt von der Ausstattung der Grafikkarte und vom entsprechenden Treiber ab. Ältere Grafikkarten können möglicherweise nur eine Farbtiefe von 16 Bit (65.536 Farben) oder gar nur 8 Bit (256 Farben) darstellen. Wenn nun ein Bild mit mehr Farben dargestellt werden soll, muss das Betriebssystem nicht vorhandene Farben durch ihre nächsten verfügbaren Nachbarn ersetzen – gerade bei nur 8 Bit Farbtiefe sieht das Ergebnis wenig realistisch aus.

Erfreulicherweise ist so gut wie jeder PC oder Mac, der in den letzten fünf bis zehn Jahren gebaut wurde, in der Lage, 16,7 Millionen Farben (24 Bit RGB) anzuzeigen; bei etwas älteren Geräten lässt sich in der Regel eine Grafikkarte nachrüsten, die diese Funktionalität liefert – beachten Sie lediglich die Anmerkungen über AGP- und PCI-Grafikkarten in Kapitel 3,

Hardware. Wenn eine Grafikkarte 24 oder gar 32 Bit Farbtiefe besitzt, wird dies als True Color bezeichnet, weil sie dadurch jede Farbe einer Bilddatei ohne Umrechnung darstellen kann.

9.1.2 Bildgröße und Auflösung

Zu den Merkmalen, die die Qualität eines Pixelbildes kennzeichnen, gehört neben der Farbtiefe auch die **Bildauflösung**. Sie bestimmt, wie viele Pixel pro Längeneinheit verwendet werden, also wie detailreich das Bild dargestellt wird. Die angegebene Längeneinheit ist meist Inch (Zoll), also 2,54 cm, manchmal aber auch Zentimeter.

Die Druckauflösung

Im Druck und bei der Bildschirmdarstellung wird die Einheit für die Auflösung ein wenig unterschiedlich bezeichnet: Auf dem Bildschirm beziehungsweise bei der Bildbearbeitung spricht man von Pixeln pro Inch (ppi) beziehungsweise Pixeln pro Zentimeter, bei der Ausgabe wird dagegen die Maßeinheit Punkte pro Inch (dpi – dots per inch) beziehungsweise Punkte pro Zentimeter eingesetzt. Der Grund dafür ist einfach: Während Pixel absolut quadratisch sind, sind Druckpunkte je nach Druckverfahren rund oder oval.

Druckraster

Darüber hinaus muss beim Drucken ein weiterer Wert berücksichtigt werden: die so genannte **Rasterweite**. Druckmaschinen, Tintenstrahl- oder Laserdrucker sind in Wirklichkeit gar nicht in der Lage, Farbe mit unterschiedlicher Intensität zu Papier zu bringen. Stattdessen drucken sie unterschiedlich große Punkte, deren Verteilung für die unterschiedlichen Helligkeitswerte sorgt.

Damit aus den vier Druckfarben die gewünschten Mischungen entstehen können, werden ihre jeweiligen Punkte nicht genau übereinander gedruckt, sondern in unterschiedlichen Winkeln. Verbreitet ist beispielsweise die folgende Rasterwinkelverteilung: Cyan 15, Magenta 75, Gelb 0 und Schwarz 45. Dies verhindert darüber hinaus die Entstehung eines Moirée-Effekts, bei dem die unpassende Überlagerung von Linien zu Darstellungsfehlern führt.

Die Rasterweite gibt den Abstand zwischen den Rasterlinien an. Sie wird in Linien pro Zentimeter (l/cm) oder Linien pro Inch (lpi) angegeben. Da neben der unterschiedlichen Größe von Rasterpunkten auch ihre Anzahl innerhalb eines Bereiches für unterschiedliche Helligkeitsabstufungen sorgen kann, werden zum Erreichen einer bestimmten Auflösung weniger Rasterlinien benötigt als erwartet. Beispielsweise verwendet der Standard-Offsetdruck 60 lpi zur Darstellung der Druckauflösung von 300 dpi. Im deutschen Sprachgebrauch bedeutet ein »60er Raster« allerdings in der Regel einen Wert von 60 l/cm – dies entspricht gut 150 lpi. Eine solche Druckqualität wird für höherwertige Printprodukte auf gestrichenem (versiegeltem, glänzendem) Papier verwendet.

Beachten Sie, dass sich die Auflösung für Printproduktionen stets auf die letztendliche Ausgabegröße auf dem Papier bezieht. Die Belichtungsfilme, die beim Offsetdruck einen wichtigen Zwischenschritt bilden, benötigen eine höhere Auflösung als der Ausdruck auf Papier, weil das Papier erheblich größer ist als der Film.

Je nach Bildart und Verwendungszweck werden unterschiedliche Druckauflösungen verwendet:

Druck-auflösungen

- Für den Standard-Vierfarbdruck gemischter Text-Bild-Vorlagen wird üblicherweise eine Druckauflösung von 300 dpi verwendet. Entsprechend werden im Bildbearbeitungsprogramm 300 ppi eingestellt, um solche Bilder vorzubereiten.
- Hochwertige Bilddrucke in Fotoqualität, beispielsweise für Hochglanzmagazine oder Bildbände, benötigen Druckauflösungen ab 600 dpi.
- Schwarzweiß- beziehungsweise Graustufendrucke, insbesondere von Strichzeichnungen, werden manchmal in noch höheren Auflösungen gedruckt, etwa mit 1200 dpi.

Die Bildschirmauflösung

Ein Problem mit der Bildschirmauflösung ergibt sich dadurch, dass Sie ein und denselben Bildschirm auf verschiedene Auflösungen einstellen können. Die meisten aktuellen Grafikkarten und Monitore unterstützen mindestens die folgenden Einstellungen:

- 640 x 480 Pixel
- 800 x 600 Pixel
- 1024 x 768 Pixel
- 1280 x 1024 Pixel

Auch diese Werte sind nicht universell gültig, weil es beispielsweise einige Notebook-Monitore oder spezielle Flatscreens mit anderen Proportionen gibt, statt den hier aufgeführten 4:3-Verhältnissen also 16:10 oder gar 16:9 unterstützen. Abgesehen davon werden bei 19-Zoll-Monitoren oder größeren Geräten noch höhere Auflösungen eingesetzt.

Es ist unmöglich, für Monitore eine bestimmte Auflösung pro Längeneinheit anzugeben. Andererseits benötigen Bilddateien aus formalen Gründen einen solchen Wert. Deshalb wird für die Bildschirmdarstellung von Bildern eine rechnerische Auflösung von 72 ppi verwendet. Mit der Realität muss dieser Wert nichts zu tun haben. Die tatsächliche Auflösung ist völlig unterschiedlich, wenn beispielsweise auf einem 15- und auf einem 19-Zoll-Monitor jeweils 1024 x 768 Pixel eingestellt werden.

Grund für 72 ppi

Für die Wahl des Wertes von 72 ppi gibt es dennoch einen guten Grund: Da 72 pt (DTP-Punkt) genau 1 Zoll (Inch) betragen, sorgt diese Auflösung dafür, dass 1 Punkt einer Schriftgröße genau einem Pixel in einem Bild entspricht. Unter Mac OS ist dies übrigens auch von Systemseite der Fall; unter Windows können Sie dagegen zwischen »großen Schriftarten« mit 72 ppi und »kleinen Schriftarten« mit 96 ppi wählen. Dennoch werden auch unter Windows stets 72 ppi für Bildschirmbilder eingestellt.

Abgesehen davon müssen Sie beim Bearbeiten von Bildern für den reinen Bildschirmeinsatz, also für Webseiten oder Multimedia-Produktionen, daran denken, dass die einzige verlässliche Maßeinheit der Pixel ist, während die Bilder für den DTP-Einsatz oft in Zentimetern oder Zoll gemessen werden, weil die absolute Ausgabegröße bekannt ist.

Die Scan-Auflösung

Eine Besonderheit müssen Sie übrigens noch beachten, wenn Sie Bilder mit einem Scanner digitalisieren. Es genügt nicht, ein Bild mit der gewünschten Bildschirm- oder Druckauflösung einzuscannen. Zum einen müssen Sie eventuell einen Skalierungsfaktor beachten: Ist das Bild beispielsweise 30 x 40 mm groß, soll aber mit einer Größe von 60 x 80 mm und einer Auflösung von 300 dpi gedruckt werden, dann müsste das Original schon rein rechnerisch mit 600 dpi gescannt werden, um die doppelte Größe bei gleich bleibender Qualität zu erreichen.

Aber auch dies genügt noch nicht: Da ein Bildpunkt mit einer bestimmten Farbe im ungünstigsten Fall genau zwischen zwei Scanlinien liegen könnte, wird in der Regel noch einmal die doppelte Auflösung verwendet, um diesen Fall auszuschließen und alle Farben und Details des ursprünglichen Bildes berücksichtigen zu können. Diese Verdoppelung der gewünschten Druckauflösung wird als **Qualitätsfaktor** bezeichnet. Ein Qualitätsfaktor von 1,5 ist gerade noch ausreichend; 2 ist optimal. Ein noch höherer Faktor bringt keine weitere Verbesserung mehr, sondern ermöglicht es Ihnen höchstens, das Bild später in einer größeren Ausgabegröße einzusetzen als ursprünglich geplant.

Das Shannon- oder Nyquist-Theorem, das dem Qualitätsfaktor zugrunde liegt, wird in Kapitel 11, *Multimedia*, im Zusammenhang mit der Digitalisierung von Sound erläutert.

Letzten Endes ergibt sich also für die Scan-Aufösung die folgende Formel:

Scan-Auflösungs-formel

Vorlagengröße x Skalierungsfaktor x Qualitätsfaktor (2)

Angenommen, ein ursprünglich 15 x 20 mm großes Bild soll mit einer Breite von 40 mm und der Druckauflösung 300 dpi ausgegeben werden. Der Skalierungsfaktor beträgt in diesem Fall 40:15, also 2,67. Die benötigte Scan-Auflösung ist somit folgende:

300 x 2,67 x 2 ≈ 1600

9.2 Bildbearbeitung mit Adobe Photoshop

In diesem Abschnitt erhalten Sie einen Überblick über die wichtigsten Funktionen des Bildbearbeitungsprogramms Adobe Photoshop CS[2]. Natürlich kann in diesem Rahmen nicht einmal ein Bruchteil der Funktionsvielfalt dieser Anwendung dargestellt werden.

Abbildung 9.1 zeigt zunächst einen Überblick über die gesamte Arbeitsoberfläche von Photoshop. Links oben sehen Sie die Werkzeugpalette mit ihren Unmengen von Auswahl-, Mal- und Vektorwerkzeugen, daneben befindet sich ein Dokumentfenster mit aktivierten Linealen und Hilfslinien, rechts sind einige wichtige Paletten zu erkennen und am oberen Rand, unmittelbar unter dem Menü, befinden sich die Werkzeugoptionen.

Nachdem Sie eine Bilddatei geöffnet, eine Vorlage eingescannt oder Bilddaten von einer Digitalkamera importiert haben, befindet sich das zu bearbeitende Bild in einem Dokumentfenster. In Photoshop können Sie beliebig viele Dokumente zur gleichen Zeit bearbeiten; die Inhalte der meisten Paletten beziehen sich dabei jeweils auf das aktivierte Dokument.

2 CS (Creative Suite) ist die gemeinsame Bezeichnung der aktuellen Adobe-Produktlinie. Numerisch handelt es sich um die Photoshop-Version 8.

12 Datei- und Datenformate

Es gibt keine größere Illusion als die Meinung, Sprache sei ein Mittel der Kommunikation zwischen Menschen.
– Elias Canetti

In den vorangegangenen Kapiteln haben Sie zahlreiche Anwendungsprogramme und ihre Arbeitsweise kennen gelernt. Dabei wurde eine Reihe von Dateiformaten verwendet, um Daten aus diesen Anwendungen für verschiedene Zwecke zu speichern. Zwar wurde an der jeweiligen Stelle kurz auf den Haupteinsatzzweck und die Besonderheiten dieser Formate eingegangen, aber näher erläutert wurden sie dort nicht. Dies geschieht zentral im vorliegenden Kapitel.

Die Wahl des richtigen Dateiformats ist unerlässlich, um effizient mit bestimmten Anwendungen zu arbeiten und um jeweils kompatible Daten für den weiteren Workflow zu erzeugen. Deshalb werden in diesem Kapitel die wichtigsten Charakteristika verschiedener Formate für Text, formatierten Text, Grafiken, Bilder und Multimediadaten behandelt.

Beachten Sie, dass dieses Kapitel die Datei- und Datenformate ausschließlich aus Anwendersicht beschreibt. Sie erfahren hier also nicht, wie Sie als Programmierer bestimmte Dateiformate lesen, schreiben und interpretieren können. Dieser spezielle Aspekt von Dateiformaten wird in dem empfehlenswerten Buch *Dateiformate – Eine Referenz* von Günther Born (Galileo Press) genau beschrieben.

12.1 Textdateien und Zeichensätze

Die einfachste Dateisorte sind einfache Textdateien, in denen jedes Zeichen sich selbst repräsentiert, in denen also keine binären Steuercodes vorkommen. Reine Textdateien werden trotzdem nicht nur zur Speicherung von Texten in menschlichen Sprachen verwendet. Es existieren auch viele textbasierte Formate, in denen Steuerungs-, Formatierungs- und Strukturbefehle im Klartext gespeichert werden, sodass sie von Menschen gelesen und geändert werden können. Wichtige Beispiele sind XML, HTML, die Satzsprache LaTeX, die Druckseitenbeschreibungssprache PostScript und nicht zuletzt die zahlreichen proprietären Konfigurationsdateien für Betriebssysteme und Anwendungen. Dies macht einen großen Teil der Bedeutung von Textdateien aus.

Sie haben in Kapitel 4, *Betriebssysteme*, erfahren, wie Textdateien mit Hilfe verschiedener System-Tools und Editoren verarbeitet werden können. In Kapitel 5, *Grundlagen der Programmierung*, wurde die programmierte Verarbeitung von Textdateien bereits kurz behandelt. Aber bisher wurde noch nicht genau darauf eingegangen, was wirklich in einer Textdatei gespeichert wird.

Formal betrachtet handelt es sich bei einer Textdatei um eine Abfolge von Speicherblöcken, in denen verschiedene Bitmuster für die unterschiedlichen Zeichen stehen. Wie breit der einzelne Speicherblock ist, hängt dabei vom verwendeten Zeichensatz ab. Beispielsweise wird der verbreitete ASCII-Zeichensatz mit seinen Erweiterungen in der Regel in

Dateien und Datenstrukturen mit 8 Bit pro Zeichen gespeichert; für Unicode stehen Varianten mit 16 oder 32 Bit oder sogar mit variabler Bit-Breite zur Verfügung.

ASCII-Code

Der ursprüngliche **ASCII**-Zeichensatz (American Standard Code for Information Interchange) wurde in den 60er-Jahren entwickelt. Er entstand durch das einfache Durchnummerieren sämtlicher Zeichen und Funktionen, die eine US-amerikanische Schreibmaschine erzeugen kann. Es gibt also nicht nur ASCII-Codierungen für Buchstaben, Ziffern und einige Sonderzeichen, sondern auch für typische Schreibmaschinenfunktionen wie den Zeilenvorschub (Line Feed), den Wagenrücklauf (Carriage Return) oder sogar die Glocke (Bell), deren Ton das baldige Zeilenende anzeigt. Diese Steuerzeichen werden von den verschiedenen Terminals, Shells oder Texteditoren unterschiedlich genutzt.

Der ASCII-Code selbst ist nur 7 Bit breit, enthält also 128 Zeichen. Da er in der Regel in 8 Bit großen Einheiten gespeichert wird, bleibt das erste Bit echter ASCII-Zeichen 0. Der verfügbare Platz von weiteren 128 Zeichen, die bei einer Speicherung in ganzen Bytes übrig bleibt, wurde zuerst von IBM als »erweiterter ASCII-Code« genutzt. In der Folge wurden die verschiedensten Zeichensätze entwickelt, die die ersten 128 Zeichen mit ASCII gemeinsam haben und die restlichen 128 für eigene Erweiterungen nutzen. Eine gängige standardisierte Fassung ist der **ANSI**-Zeichensatz, der beispielsweise in den Windows-Versionen der meisten lateinisch schreibenden europäischen Sprachen zum Einsatz kommt. Er enthält die wichtigsten west- und mitteleuropäischen Sonderzeichen wie etwa deutsche Umlaute oder französische Buchstaben mit Akzent. ANSI steht für American National Standards Institute, es handelt sich um das US-Pendant zum deutschen DIN oder zur internationalen ISO.

12.1.1 Das Problem des Zeilenumbruchs

Das einzige Sonderzeichen, das in jeder Textdatei geduldet wird und eine universelle Bedeutung besitzt, ist der Zeilenumbruch. Er entsteht bei der manuellen Eingabe durch das Drücken der Eingabetaste (Return oder Enter). Ärgerlicherweise wird er selbst in ASCII-Textdateien für die verschiedenen Plattformen durch unterschiedliche Zeichen dargestellt:

- UNIX-Varianten wie Linux, FreeBSD und Solaris verwenden ein einzelnes **LineFeed** (LF, ASCII-Code 10).
- MacOS benutzt traditionellerweise ein einzelnes **Carriage Return** (CR, ASCII-Code 13). Allerdings verwendet die Darwin-Umgebung in MacOS X das LF, da es sich um ein UNIX handelt.
- Windows und MS-DOS verwenden eine Kombination aus beiden Zeichen: ein CR, gefolgt von einem LF (in der Regel als CRLF bezeichnet). Diese Zeichenfolge wird übrigens auch in der Netzwerkkommunikation von vielen Protokollen eingesetzt (siehe Kapitel 13, *Netzwerktechnik*).

Die Verwendung unterschiedlicher Zeichen für Zeilenumbrüche führt immer wieder zu Problemen bei der Konvertierung oder beim Öffnen von Textdateien, die unter einem anderen Betriebssystem gespeichert wurden. Beispielsweise sehen Sie beim Öffnen einer Windows-Textdatei unter dem klassischen MacOS ein kleines Quadrat (als nicht darstell-

bares Sonderzeichen) am Anfang jeder neuen Zeile, während Sie unter Windows nur kleine Quadrate anstelle der Zeilenumbrüche sehen, wenn Sie eine Mac- oder UNIX-Textdatei öffnen.

Konvertierung

Dieses Problem lässt sich durch intelligente Editoren in den Griff bekommen, beispielsweise erkennt TextPad für Windows (Demoversion unter **www.textpad.com**) die Fremdformate automatisch und stellt die Dateien intern um. Auch der in MacOS X eingebaute GUI-Texteditor TextEdit kann sich auf Windows-Zeilenendungen einstellen. Unter Linux sind spezielle Konverterprogramme für die Konsole verfügbar: `fromdos Dateiname` konvertiert die angegebene Datei mit Windows-Zeilenumbrüchen auf UNIX-Umbrüche; `todos Dateiname` erledigt die umgekehrte Aufgabe. Für Mac- und Windows-Benutzer ist zusätzlich das Programm AscOnA nützlich, weil es nicht nur die Zeilenendungen in beide Richtungen konvertiert, sondern auch gleich die Umlaute und andere diakritische Zeichen dem jeweiligen Zeichensatz anpasst. Die Software kann für beide Plattformen kostenlos unter der URL **http://www.medienwerkstatt-online.de/products/ascona/ascona.html** heruntergeladen werden.

Wenn Sie innerhalb von Anwendungsprogrammen mit Text arbeiten, haben Sie übrigens kein Problem mit Zeilenumbrüchen: Layoutsoftware wie InDesign und QuarkXPress oder Textverarbeitungsprogramme wie Microsoft Word, die für Windows und MacOS verfügbar sind, speichern auch die enthaltenen Texte in ihrem eigenen Format, in dem die Zeilenumbrüche speziell geschützt werden. Dabei spielt es auch keine Rolle, ob das jeweilige Programm ein binäres oder XML-basiertes Dateiformat verwendet.

Auch Programmiersprachen beschäftigen sich mit dem Problem der unterschiedlichen Zeilenumbrüche. Wenn Sie in den Sprachen, die in diesem Buch näher behandelt werden, also C, Java oder Perl, die Escape-Sequenz `\n` ausgeben, gilt sie als logischer Zeilenumbruch, es wird also hinter den Kulissen das jeweils passende Zeichen beziehungsweise die korrekte Zeichenfolge erzeugt. Dennoch steht `\n` eigentlich für das LF, also das Zeichen mit dem ASCII-Code 10, während `\r` das CR (ASCII-Code 13) repräsentiert.

Perl-Variable $/

Die Programmiersprache Perl geht noch einen Schritt weiter: Sie definiert eine spezielle Variable namens `$/`, den Input Record Separator. Diese enthält das jeweils aktuelle Zeichen oder die Zeichenfolge, die als Zeilentrenner für den Operator <> gilt. Wenn Sie eine Textdatei erhalten, die von einer anderen Plattform stammt, können Sie den Wert der Variablen einfach überschreiben: Entweder sichern Sie ihren Wert in einer zweiten Variablen oder erstellen sich mit Hilfe des Schlüsselworts `local` eine gleichnamige lokale Variable, die die »offizielle« Variable innerhalb des aktuellen Gültigkeitsbereichs ausblendet. Wenn Sie `$/` einen neuen Wert zuweisen, müssen Sie allerdings darauf achten, dass Sie die Zeichen CR und LF nicht einfach durch `\r` beziehungsweise `\n` (den logischen Umbruch) darstellen dürfen, sondern über ihre ASCII-Codes ansprechen müssen. Dies geschieht in Perl (genau wie in C oder Java) durch einen Backslash mit dem nachstehenden oktalen Zeichencode: `\012` steht für LF, `\015` stellt CR dar.

Das folgende kurze Beispiel dient dem zeilenweisen Einlesen einer klassischen Mac-Textdatei auf einer beliebigen Plattform:

```
local $/ = "\015";
open (MAC, "<mactextfile.txt") || die "Geht nicht!\n";
```

```
while ($line = <MAC>) {
   chomp $line;
   push @text, $line;
}
close MAC;
```

Sie können in `$/` übrigens auch einfach einen leeren String speichern, um eine Textdatei vollständig auf einmal zu »saugen«:

```
my $umbr = $/;          # alten Wert schützen
$/ = "";                # "Saug"-Modus
open (FILE, "<test.txt") || die "Geht nicht!\n";
my $text = <FILE>;      # alles auf einmal lesen
close FILE;
$/ = $umbr;             # $/ wiederherstellen
```

12.1.2 Zeichensätze

In diesem Unterabschnitt werden die Grundlagen einiger Zeichensätze behandelt – vom einfachen ASCII- bis zum komplexen Unicode-Zeichensatz.

ASCII

Der ASCII-Zeichensatz (gesprochen »äskie«) ist der wichtigste und grundlegendste aller Computerzeichensätze. Viele klassische Konsolen- oder Netzwerkanwendungen verarbeiten keine Zeichen über reines ASCII hinaus; Bezeichner in traditionellen Programmiersprachen wie C dürfen ebenfalls nur aus ASCII-Zeichen bestehen.

Der ASCII-Zeichensatz geht auf einen ANSI-Vorschlag von 1963 zurück und wurde im Wesentlichen bei IBM ausgearbeitet, vor allem von **Robert Bemer** (**www.bobbemer.com**), der unter anderem auch einen entscheidenden Beitrag zur Entwicklung der Programmiersprache COBOL geleistet hat. 1968 wurde der ASCII-Code standardisiert.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	»	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Tabelle 12.1 Der ASCII-Zeichensatz

Tabelle 12.1 zeigt zunächst den gesamten ASCII-Zeichensatz. Die Nummerierung ist hexadezimal; in der jeweiligen Zeile ist die hohe (Sechzehner-)Stelle eines Codes zu sehen, in der Spalte die niedrige (Einer-)Stelle. Das Zeichen `'A'` besitzt also zum Beispiel den Code 41 (dezimal 65).

Die Zeichen 0 bis 31 (hexadezimal 00 bis 1F) sind Steuerzeichen, die mit ihren traditionellen Kurznamen verzeichnet wurden. Beispielsweise ist 0A (dezimal 10) das bereits erwähnte LineFeed, 0D (13) ist das Carriage Return und 00 (das Zeichen `'\0'`) wird vor allem als Abschlussmarkierung für C-Strings verwendet.

Wie Sie sehen, haben die Entwickler sich bemüht, bestimmte Zeichenklassen an charakteristischen Stellen unterzubringen. Außerdem kommen die Ziffern und die beiden Sätze von Buchstaben in ihrer natürlichen Reihenfolge vor: Die Ziffernzeichen `'0'` bis `'9'` befinden sich auf den Positionen 48 bis 57 (hexadezimal 30 bis 39), die Großbuchstaben `'A'` bis `'Z'` auf 65 bis 90 (hexadezimal 41 bis 5A) und die Kleinbuchstaben `'a'` bis `'z'` auf 97 bis 122 (hexadezimal 61 bis 7A).

ASCII-Erweiterungen

In den 80er-Jahren war es so gut wie selbstverständlich, auf einem Home oder Personal Computer keine deutschen Umlaute und sonstigen Sonderzeichen zur Verfügung zu haben: ASCII ist nur für englische Texte geeignet, alle anderen Sprachen mussten ihre Sonderzeichen umschreiben (`ae` für `ä` und so weiter) oder spezielle diakritische Kennzeichnungen einfach weglassen (zum Beispiel Französisch ohne Akzente).

Grenzen von ASCII

Da Computer jedoch ab einem bestimmten Zeitpunkt im großen Stil Einzug in Firmen und Behörden hielten und die Schreibmaschine als Korrespondenzwerkzeug zunehmend ablösten, war dieser Zustand irgendwann nicht mehr erträglich: Deutscher Text ohne Umlaute und ähnliche falsche Schreibweisen anderer Sprachen sind viel schlechter lesbar als richtig geschriebener Text. Da niemand seinen Kunden auf Dauer solche Notlösungen in Geschäftsbriefen zumuten wollte, mussten die fehlenden Zeichen integriert werden.

Erfreulicherweise verwenden alle heutigen Computer Bytes, die acht Bits groß sind (das war in der Frühzeit der Computergeschichte nicht immer selbstverständlich; der oben genannte ASCII-Entwickler Bob Bemer war übrigens einer derjenigen, die sich dafür eingesetzt hatten). Aus diesem Grund war es weit verbreitete Praxis, jedes einzelne eigentlich nur 7 Bits breite ASCII-Zeichen in einem 8 Bits großen Block zu speichern. In einem Byte war also Platz für weitere 128 Zeichen, die man irgendwie ausnutzen konnte.

IBM war das erste Unternehmen, das einen solchen Zeichensatz einführte – der so genannte »erweiterte ASCII-Zeichensatz« enthielt die wichtigsten diakritischen Zeichen west- und mitteleuropäischer Sprachen sowie eine Reihe von geraden und gewinkelten Linien und anderen Grafikzeichen, um konsolenbasierte Benutzeroberflächen realisieren zu können. Dieser Zeichensatz wurde beispielsweise in leicht abgewandelter Form von MS-DOS verwendet.

Andere Hersteller erfanden ihre eigenen erweiterten Zeichensätze. Zum Beispiel entschied sich Apple für eine Eigenentwicklung, in der die 128 Zeichen über ASCII anders belegt wurden. Deswegen können Sie bis heute eine deutsche Mac-Textdatei nur mit Fehlern unter Windows lesen und umgekehrt.

ANSI-Zeichensatz

Apropos Windows: Microsoft entschloss sich, für Windows nicht IBMs erweiterten ASCII-Zeichensatz oder die MS-DOS-Variante davon zu verwenden, sondern den inzwischen standardisierten ANSI-Zeichensatz: Da Windows eine grafische Benutzeroberfläche war, wurden die verschiedenen Grafikzeichen nicht mehr gebraucht, um Fenster oder Menüs zu zeichnen. Den frei gewordenen Platz benutzt der ANSI-Zeichensatz für mehr internationale diakritische Zeichen. Das ist der Grund dafür, warum ein Konsolenprogramm, das Sie in einem Windows-Editor schreiben, statt deutscher Umlaute seltsame Sonderzeichen anzeigt. Die von DOS übernommene Windows-Konsole verwendet nach wie vor den alten MS-DOS-Zeichensatz!

Als Computer sich in immer mehr Ländern zu verbreiten begannen, fiel unangenehm auf, dass die ANSI-Zeichen noch nicht einmal für alle lateinisch geschriebenen Sprachen geeignet waren. Beispielsweise benutzen Sprachen wie Tschechisch oder Türkisch einige besondere Zeichen, um spezielle Laute zu kennzeichnen.

Codepages

Die Betriebssystemhersteller begegneten diesem Problem, indem sie für jedes Land, in dem das entsprechende System verkauft wurde, einen angepassten Zeichensatz oder eine so genannte **Codepage** herstellten, um die Besonderheiten der jeweiligen Sprache darstellen zu können. Neben den verschiedenen Varianten der lateinischen Schrift wurden so auch andere Buchstabenalphabete realisiert, beispielsweise Kyrillisch, Griechisch, Hebräisch oder Arabisch. Für die beiden Letztgenannten musste natürlich zusätzlich das Problem gelöst werden, dass sie von rechts nach links geschrieben werden. Für sich allein genommen ist das noch nicht einmal schwierig, vielmehr bereitet die Mischung mit englischem Text (also der umgekehrten Schreibrichtung), der für die Programmierung und Ähnliches eingesetzt wird, Kopfzerbrechen.

ISO 8859

Mit den betriebssystemabhängigen Lösungen waren einige Jahre lang fast alle recht zufrieden; Anwendungsprogramme, die für MacOS und Windows verfügbar waren, verwendeten ohnehin ihre eigenen inneren Mechanismen für den Erhalt des korrekten Zeichensatzes. Zum Problem wurden diese Insellösungen erst, als sich das Internet immer schneller ausbreitete. Deshalb wurde für plattformübergreifende Kompatibilität ein international gültiger Standard für Zeichensätze geschaffen: der ISO-8859. Die wichtigsten ISO-8859-Zeichensätze werden in Tabelle 12.2 gezeigt.

ISO-Zeichensatz	Alternativer Name	Beschreibung
ISO-8859–1	ISO-Latin-1	Entspricht weitgehend dem ANSI-Zeichensatz. Unterstützt die Sprachen von USA und Westeuropa.
ISO-8859–2	ISO-Latin-2	Unterstützt zusätzlich einige slawische Sprachen.
ISO-8859–3	ISO-Latin-3	Unterstützt Esperanto, Maltesisch und Türkisch.
ISO-8859–4	ISO-Latin-4	Dient der Unterstützung der baltischen Sprachen (Estnisch, Lettisch und Litauisch).
ISO-8859–5	–	Kyrillisch
ISO-8859–6	–	Arabisch

Tabelle 12.2 Die wichtigsten standardisierten ASCII-Erweiterungs-Zeichensätze

ISO-Zeichensatz	Alternativer Name	Beschreibung
ISO-8859–7	–	Griechisch
ISO-8859–8	–	Hebräisch
ISO-8859–9	ISO-Latin-5	Wie Latin-1; ersetzt aber isländische durch türkische Zeichen. Wird für Türkisch häufiger verwendet als Latin-3.
ISO-8859–10	ISO-Latin-6	Unterstützt Grönländisch und Lappisch.

Tabelle 12.2 Die wichtigsten standardisierten ASCII-Erweiterungs-Zeichensätze (Forts.)

Beachten Sie, dass es noch weitaus mehr angepasste Zeichensätze gibt als die hier dargestellten. Nicht »totzukriegen« sind beispielsweise die traditionellen Mac-Zeichensätze.

Unicode

Um das ganze Problem der Schriften ein für alle Mal in den Griff zu bekommen, wurde schließlich das **Unicode**-System erfunden. Hier wurde von vornherein mehr Speicherplatz für das einzelne Zeichen eingeplant, um mehr unterschiedliche Zeichen darstellen zu können. Die ursprünglichen Unicode-Versionen verwendeten 16 Bit pro Zeichen, um bis zu 65.536 Zeichen darstellen zu können; seit Unicode 3.1 gibt es auch spezielle Varianten mit 3 oder 4 Byte, um noch erheblich mehr Zeichen in Unicode unterzubringen.

Das größte Problem, das Unicode mit sicht bringt, ist, dass das System bisher noch nicht von allen Betriebssystemen, Programmiersprachen, APIs und Anwendungen verwendet wird. Um dieses Problem zu verringern, wurde Unicode so konstruiert, dass die untersten 256 Zeichen genau dem Zeichensatz ISO-Latin-1 entsprechen.

Die spezielle Unicode-Codierung UTF-8 kann ASCII-Zeichen sogar einfach in einem Byte darstellen und verwendet spezielle Codes zur Darstellung höherer Zeichen. Auf diese Weise ist UTF-8 zu ASCII abwärts kompatibel.

16-Bit-BMP

Die ursprüngliche Teilmenge von Unicode, die 16 Bit breite Zeichen verwendet, wird als Basic Multilingual Pane (BMP) bezeichnet. Es handelt sich um die wichtigsten Zeichen lebender Sprachen, mathematische Sonderzeichen sowie grundlegende Piktogramme, Schmuckelemente und andere Symbole (Dingbats). In Tabelle 12.3 sehen Sie eine Übersicht über einige wichtige Codebereiche des BMP-Bereichs von Unicode. Die Codes von Unicode-Zeichen werden üblicherweise im Hexadezimalcode mit vorangestelltem U+ geschrieben; beim BMP-Bereich werden vier Hexadezimalstellen verwendet, ansonsten je nach Bedarf mehr.

Hexadezimal	Dezimal	Codebereich
U+0000 – U+007F	0 – 127	ASCII
U+0080 – U+00FF	128 – 255	ISO-Latin-1

Tabelle 12.3 Einige wichtige Codebereiche im Unicode-Block BMP

Hexadezimal	Dezimal	Codebereich
U+0100 – U+017F	256 – 383	Latin Extended A (speziellere diakritische Zeichen)
U+0180 – U+024F	384 – 591	Latin Extended B (spezielle, eigentlich nicht lateinische Zeichen, die in einigen Sprachen verwendet werden)
U+0370 – U+03FF	880 – 1023	Griechisch
U+0400 – U+04FF	1024 – 1279	Kyrillisch
U+0590 – U+05FF	1424 – 1535	Hebräisch
U+0600 – U+06FF	1536 – 1791	Arabisch
U+0900 – U+097F	2304 – 2431	Devanagari (Schrift der meisten indischen Sprachen)
U+2200 – U+22FF	8704 – 8959	Mathematische Symbole
U+3040 – U+309F	12352 – 12447	Hiragana (japanische Lautschrift)
U+30A0 – U+30FF	12448 – 12543	Katakana (japanische Lautschrift)
U+4E00 – U+9FA5	19968 – 40869	CJK Unified Ideographs (chinesische Zeichen, die z. T. auch für Japanisch und Koreanisch verwendet werden)

Tabelle 12.3 Einige wichtige Codebereiche im Unicode-Block BMP (Forts.)

Eine Übersicht über alle Zeichenbereiche von Unicode erhalten Sie auf der Website des Unicode-Konsortiums: **http://www.unicode.org**. Dort finden Sie außerdem PDF-Dokumente zum Download, in denen jedes einzelne Zeichen zu sehen ist.

Unicode im Einsatz

Unicode wird inzwischen von den meisten Betriebssystemen unterstützt. Den Anfang machte Windows NT 4.0, das 1996 auf den Markt kam und den damaligen 16-Bit-Unicode als festgelegtes Textformat verwendete. Inzwischen sind auch neuere UNIX- und Linux-Versionen mit Unicode-Unterstützung ausgestattet.

Bei den Programmiersprachen machte Java den Anfang: die Sprache verwendete von Anfang an Unicode in verschiedenen Codierungen. Der Java-Datentyp `char` (einzelnes Zeichen) besitzt eine Breite von 16 Bit und ist somit groß genug, um ein BMP-Unicode-Zeichen aufzunehmen. In C wurde dazu nachträglich der 16 Bit breite Datentyp `wchar_t` eingeführt. Perl unterstützt Unicode erst seit der aktuellsten Version 5.8.

Auch immer mehr Anwendungsprogramme unterstützen Unicode. Wichtig ist dies besonders für Layout- und Textverarbeitungsprogramme. Das in Kapitel 10, *DTP und digitale Druckvorstufe*, vorgestellte InDesign arbeitet beispielsweise von Hause aus mit Unicode.

Die Unicode-Verwendung in Anwendungsprogrammen geht einher mit der Unterstützung des neuen OpenType-Schriftformats, das ebenfalls in Kapitel 10 angesprochen wurde.

Schrift- und Tastatureinstellungen

Die in den vorigen Abschnitten besprochene theoretische Unterstützung verschiedener lokaler Zeichensätze oder des Unicode-Systems nutzt in der Praxis natürlich nur dann etwas, wenn auf dem verwendeten Rechner entsprechende Schriften installiert sind und die Tastaturbelegung so geändert werden kann, dass die speziellen Zeichen anderer Schriftsysteme über die Tastatur eingegeben werden können.

Die reine Anzeige fremder Schriften ist besonders für Webseiten wichtig. Folgerichtig bieten Webbrowser wie der Microsoft Internet Explorer ein relativ einfaches System zum Umstellen der Zeichencodierung und zum Nachladen verschiedenster Zeichensätze an. Diese Darstellung anderer Zeichensätze ermöglicht aber noch lange nicht deren Eingabe!

Beispiel Windows

Die meisten aktuellen Betriebssysteme bieten eine einfache Möglichkeit, das Tastaturlayout auf viele verschiedene Schriften umzustellen und entsprechende Zeichensätze zu installieren. Unter Windows XP nehmen Sie die Einstellung der auswählbaren Tastaturlayouts beispielsweise folgendermaßen vor:

1. Wählen Sie **Start · Systemsteuerung · Regions- und Spracheinstellungen**.
2. Wechseln Sie auf die Registerkarte **Sprachen**. Über den Button **Details** können Sie die eigentlichen Einstellungen für Tastatur-Eingabeschemata vornehmen.
3. Mit **Hinzufügen** können Sie zahlreiche Sprachen und Eingabeschemata installieren.
4. Nachdem Sie die Auswahl bestätigt haben, steht Ihnen in der Taskleiste ein Umschalter zur Verfügung, mit dessen Hilfe Sie die verschiedenen Sprachen und Gebietsschemata nach Länderkürzel aussuchen können. Moderne Anwendungen wie Word oder InDesign suchen beim Umschalten in der Regel auch gleich eine Schrift aus, die die benötigten Zeichen darstellen kann. Auch das Umschalten der Schreibrichtung für verschiedene Sprachen funktioniert normalerweise problemlos.

Unter MacOS X steht Ihnen ein sehr ähnliches Verfahren zur Verfügung; hier finden Sie die erforderlichen Einstellungen unter **Systemeinstellungen · Landeseinstellungen**.

Tastenbelegung

Nun müssen Sie nur noch herausfinden, mit welcher Taste welches Zeichen einer Sprache produziert wird. Bei lateinisch geschriebenen Sprachen kann die Belegung geringfügig (Z und Y sowie einige Satzzeichen vertauscht bei englischer Tastaturbelegung), aber auch erheblich von der deutschen abweichen (zum Beispiel AZERTY statt QWERTZ bei französischem Tastaturlayout). Für nichtlateinische Schriften gelten dagegen völlig andere Regeln: Fremde Buchstabenschriften (Griechisch, Arabisch und so weiter) sind natürlich *irgendwie* auf die Tasten der Tastatur verteilt.

Viel komplexer wird es dagegen bei den ostasiatischen Silbenschriften: In der Regel werden chinesische oder japanische Zeichen anhand einer Lateinumschrift eingegeben, Sie können sich das passende Zeichen daraufhin aus einer Liste aussuchen. Bestimmte Zeichen erkennt ein solches Eingabesystem sogar bereits aus dem Kontext richtig.

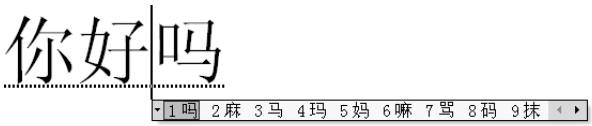

Abbildung 12.1 Eingabe chinesischer Zeichen unter Windows XP

Ein Beispiel: Chinesisch schreiben auf lateinischen Tastaturen

In Abbildung 12.1 sehen Sie ein Beispiel für die Eingabe chinesischer Schriftzeichen unter Windows XP. Die chinesische Sprache kennt viele tausend Silbenzeichen, aber nur verhältnismäßig wenige unterschiedliche Sprechsilben. Erschwerend kommt hinzu, dass es vier verschieden gesprochene, bedeutungsunterscheidende Betonungen gibt, die bei den üblichen Eingabesystemen nicht mit eingegeben werden.

In dem Beispiel sind die vereinfachten chinesischen Schriftzeichen zu sehen, die in der Volksrepublik aufgrund einer Schriftreform eingeführt wurden (Taiwan verwendet dagegen bis heute die noch komplexeren traditionellen Schriftzeichen). Die Lateinumschrift wurde ebenfalls vereinheitlicht; die chinesische Regierung hat vor Jahren ein verbindliches System namens Hanyu Pinyin eingeführt. Dies ersetzt die zum Teil abenteuerlichen Versuche portugiesischer Jesuitenpater und englischer Seefahrer; statt »Peking« heißt es beispielsweise korrekt »Beijing«.

Um die obigen Zeichen zu produzieren, müssen Sie die folgenden Silben eingeben: NI HAO MA (das Leerzeichen schließt jeweils die Eingabe einer Silbe ab). Da dieser Satz »Wie geht es dir?« bedeutet und sehr häufig vorkommt, hat das System auf Anhieb die korrekten Zeichen ausgewählt. Wird ein Zeichen dagegen falsch erkannt, können Sie einmal auf die linke Pfeiltaste drücken, um die im Bild gezeigte Auswahlleiste zu erhalten. Bei den meisten Silben besteht sie aus mehreren Abschnitten, die Sie durch die Pfeil-Buttons rechts erreichen können. Unter Mac OS X ist der Mechanismus sehr ähnlich.

12.1.3 Textbasierte Dateiformate

Viele wichtige Dateiformate, die in reinem Text geschrieben werden, lernen Sie in anderen Kapiteln dieses Buches kennen: In Kapitel 15 wird beispielsweise ausführlich die allgemeine Auszeichnungssprache XML behandelt; in Kapitel 16 kommt die Webseitenbeschreibungssprache HTML an die Reihe. Auch die verschiedenen Programmiersprachen und diverse Arten von Konfigurationsdateien lassen sich letzten Endes als Textdateiformate bezeichnen.

An dieser Stelle werden deshalb nur zwei Beispiele für textbasierte Formate angeführt: die Satzsprache LaTeX und die Druckseitenbeschreibungssprache PostScript. Ein drittes Beispiel ist das von Microsoft definierte Rich Text Format (RTF), das als rudimentäres Austauschformat für formatierten Text zwischen verschiedenen Anwendungen eingesetzt werden kann.

LaTeX

Die WYSIWYG-Dokumentenverarbeitung im Bereich des Textsatzes war nicht zu allen Zeiten selbstverständlich. Fotosatzmaschinen und frühe computerbasierte Satz- und Layoutsysteme verwendeten angepasste Spezialsprachen, um das Layout und den Textinhalt von Druckdokumenten zu beschreiben.

TeX

Eine der bekanntesten Textsatzsprachen wurde in den 80er-Jahren von Donald E. Knuth[1] entworfen. Hauptberuflich war er ein sehr bekannter Informatikprofessor – sein Hauptwerk ist das bisher dreibändige, auf sieben Bände ausgelegte »The Art of Computer Programming«. 1982 gab er das Manuskript des ersten Bandes beim Verlag ab und war enttäuscht über den schlechten Satz, der dabei herauskam. Dies veranlasste ihn, mit der Arbeit an seinem eigenen Textsatzsystem zu beginnen, genannt TeX (gesprochen »Tech« – mit ch wie in »Blech«).

Zusammen mit TeX entwickelte Knuth ein weiteres Programm namens METAFONT zur Definition vektorbasierter elektronischer Schriften. Die wichtigsten Ideen von METAFONT führten später zur Entwicklung der Adobe-PostScript-Schriften. In TeX gesetzte Dokumente bestechen durch ihre hohe Qualität, die nicht einmal von professionellen Layoutprogrammen wie InDesign oder QuarkXPress erreicht wird, sondern höchstens von der Kunst des manuellen Bleisatzes. Außerdem ist kein anderes Programm in der Lage, in ähnlicher Perfektion mathematische Formeln zu setzen.

Andererseits ist TeX schwer zu benutzen; es müssen Unmengen von Konfigurationsanweisungen geschrieben werden, bevor das eigentliche Dokument gesetzt werden kann. Aus diesem Grund fügten andere Entwickler eine Makrosammlung namens LaTeX hinzu, die viele der Konfigurationsarbeiten durch Zusammenfassung erleichtert, sodass Autoren sich mehr auf den eigentlichen Text konzentrieren können als auf die Satzsprache.

Eine richtige Einführung in LaTeX würde an dieser Stelle erheblich zu weit führen; deshalb soll ein kurzes Beispieldokument genügen:

```
\begin{document}
\textbf{\Large LaTeX}
LaTeX ist eine Makrosammlung, die auf der Satzsprache Tex von \
emph{D.E.Knuth}\footnote{Der Autor der bekannten Art Of Computer Pro-
gramming.} basiert. LaTeX bietet eine Reihe von Vorteilen:
\begin{itemize}
\item
Gestochen scharfer Textsatz.
\item
Mathematische Formeln in Perfektion.
\item
Listen, Tabellen und Abbildungen ohne Probleme.
\end{itemize}
\end{document}
```

1 Der Name wirk »K-nooth« (englisch) ausgesprochen.

Wie Sie sehen, werden Formatierungsbefehle mit `\befehl` eingeleitet; die Inhalte, die ein Befehl umschließt, stehen in geschweiften Klammern. Der obige Beispielcode würde etwa folgende Ausgabe erzeugen:

LaTeX

LaTeX ist eine Makrosammlung, die auf der Satzsprache Tex von **D.E.Knuth** basiert. LaTeX bietet eine Reihe von Vorteilen:

- Gestochen scharfer Textsatz.
- Mathematische Formeln in Perfektion.
- Listen, Tabellen und Abbildungen ohne Probleme.

LaTeX verarbeiten

Ein solches Dokument erhält die Dateiendung `.tex` und wird vom eigentlichen LaTeX-Programm in eine DVI-Datei (Device-Independent Print File) umgewandelt, die über das Programm `dvips` in PostScript umgewandelt werden kann. Die freie PostScript-Implementierung GhostScript ermöglicht schließlich die Vorschau und Ausgabe des Dokuments. Auf einem UNIX-System würden Sie dazu beispielsweise folgende Befehle eingeben:

```
$ latex latexinfo.tex
$ dvips latexinfo.dvi
$ gs latexinfo.ps
```

Heutzutage gibt es auch einfache Möglichkeiten, direkt aus LaTeX-Dokumenten das modernere PDF-Format zu generieren.

PostScript

Die von Adobe eingeführte Druckseitenbeschreibungssprache PostScript wurde für die millimetergenaue Platzierung von Text, Vektorgrafik und Bildern erfunden, die für die Ausgabe auf Laserdruckern und Belichtern erforderlich ist. Die Sprache verfügt über eingebaute Funktionen zur Formatierung von Text und zur Anfertigung von Vektorzeichnungen; Pixelbilder werden dagegen mit der PostScript-Datei verknüpft.

Es handelt sich zwar um ein textbasiertes Format, aber in aller Regel gibt es niemand von Hand ein. PostScript-Dokumente werden fast immer von entsprechenden Druckertreibern erzeugt. Eine PPD (Druckerbeschreibungsdatei) wird dabei verwendet, um die speziellen Fähigkeiten des verwendeten Druckers wie Auflösung, Farbfähigkeit und eingebaute Schriften zu bestimmen. Aus diesen Gründen würde es auch keinen Sinn ergeben, hier PostScript-Code darzustellen.

Wenn Sie ein PostScript-Dokument als Datei speichern möchten, müssen Sie Ihren Druckertreiber anweisen, die Druckdaten an die Datei statt an den Drucker selbst zu senden. Das Verfahren wurde bereits in Kapitel 10, *DTP und digitale Druckvorstufe*, kurz erläutert.

EPS

Eine spezielle Variante von PostScript ist das EPS-Format (Encapsulated PostScript). Es wird eher zu den Bilddateiformaten gerechnet, da es von den meisten Grafik- und Bildbearbeitungsprogrammen exportiert und geöffnet werden kann. EPS-Dateien sind die bevorzugte

Methode, um Vektorgrafiken oder gemischte Pixel-/Vektorgrafiken in Layoutdokumente zu importieren. Gegenüber dem klassischen PostScript-Format weist EPS die folgenden Besonderheiten auf:

- Die Konturen von Schriftarten können ähnlich wie bei PDF mit in das Dokument verpackt werden.
- Pixelbilder werden nicht verknüpft, sondern ihr Datenbestand wird grundsätzlich in das EPS eingebettet.
- EPS-Dokumente sind auf eine einzige Seite beschränkt, während PostScript beliebig viele Seiten einnehmen kann.

12.2 Binäre Dateiformate

Einen grundsätzlich anderen Weg als die bisher besprochenen textbasierten Dateiformate gehen die **Binärformate**. Es handelt sich um Dateien mit numerischen Inhalten, wobei ohne Kenntnis der Spezifikation des jeweiligen Dateiformats nicht erkennbar ist, ob die gespeicherten Zahlen an bestimmten Stellen für Befehle oder für Nutzdaten stehen. Viele Formate nehmen nichtsdestotrotz eine klare Trennung vor: Sie schreiben die Verwaltungsinformationen zu Beginn der Datei in einen Header und hängen die Nutzdaten hinten an. Auf diese Weise wird einem Programm, das eine solche Datei öffnet, zuerst mitgeteilt, wie genau es mit den Daten verfahren soll, bevor diese Daten selbst gelesen werden.

Wenn Sie Binärdateien in einem Texteditor öffnen, bekommen Sie ein einziges Chaos seltsamer Sonderzeichen zu sehen, weil die Binärdaten irrtümlich als Zeichen eines Zeichensatzes interpretiert werden. Sollten Sie eine Binärdatei aus dem Texteditor heraus speichern, geht sie ziemlich sicher kaputt – der Editor wird versuchen, seine für Textdokumente gedachte Zeilenumbruchlogik auf die Binärdaten anzuwenden, wodurch sich die Bytes des Dokuments wahrscheinlich verschieben.

Wenn Sie eine Binärdatei überhaupt manuell und nicht mit dem zuständigen Programm öffnen möchten, können Sie das nur mit einem Hexadezimal-Editor tun, der die einzelnen Bytes des Dokuments als Hexwerte und manchmal zusätzlich als Text anzeigt. Abbildung 12.2 zeigt den Anfang einer TIFF-Bilddatei im Hex-Modus des Windows-Editors TextPad.

Dass die Zeichendarstellung der Binärwerte rechts neben den Hexadezimalkolonnen zum Teil lesbaren Text enthält, ist übrigens kein Widerspruch zu der Aussage, dass Binärdokumente nur numerische Werte enthalten: Zufälligerweise wird Text, der im Dokument selbst oder – wie hier – für Verwaltungszwecke eingesetzt wird, numerisch in einem bestimmten Zeichencode abgelegt. Dies ist eine von vielen Varianten zur Speicherung von Text innerhalb von Binärdokumenten, die eben auch vielerlei andere Arten von Daten enthalten können.

Beachten Sie, dass Binärdaten als solche plattformabhängig sind. Es bestehen zwei verschiedene große Probleme:

- Verschiedene Plattformen verwenden unterschiedliche Wortbreiten wie 16, 32 oder 64 Bit.
- Es gibt zwei verschiedene Reihenfolgen für die Darstellung von Werten, die mehrere Bytes breit sind: So genannte Big-Endian-Architekturen wie Motorola 68K oder

16.2 Cascading Stylesheets (CSS)

CSS sind kein integraler Bestandteil von HTML, gehören aber zu den Neuerungen, die das W3C bereits seit der HTML-4.0-Spezifikation empfiehlt und fördert. Durch Stylesheets wird die Struktur eines HTML-Dokuments vollständig von seinem Inhalt getrennt. Dies macht die Dokumente einerseits erheblich übersichtlicher, weil die unzähligen layoutorientierten Tags entfallen, und ermöglicht zum anderen die saubere Definition des Layouts an zentraler Stelle.

Konkret geht es darum, Stilvorlagen für das Aussehen beliebiger HTML-Elemente zu definieren. Das bedeutet, dass Sie mit Hilfe von CSS sehr konkret in das Layout einer Webseite oder einer ganzen Site eingreifen können, indem Sie etwa festlegen, dass alle Überschriften vom Typ `<h1> ... </h1>` rot, zentriert und in kursiver 36-Punkt-Helvetica erscheinen sollen.

Die Arbeit mit Stylesheets hat folgende Vorteile:

- Die Erzeugung eines immer wiederkehrenden, einheitlichen Layouts wird erheblich vereinfacht – einmal festgelegt, können bestimmte Formatierungen automatisch wiederholt werden.
- Es ist mit Hilfe von Stylesheets erstmals möglich, Elemente auf einer Webseite pixelgenau zu platzieren – im Extremfall sogar übereinander. Auf diese Weise können Sie beispielsweise Text mit Schatten oder Text auf einem Bild realisieren.
- In Zusammenarbeit mit JavaScript können Elemente, die CSS zur Stilfestlegung verwenden, dynamisch in ihren Eigenschaften verändert werden. In diesem Zusammenhang spricht man gern von DHTML (Dynamic HTML). Das W3C empfiehlt dazu die Verwendung eines allgemeinen Objektmodells, das alle Elemente des Browsers und des HTML-Dokuments standardisiert (DOM, Document Object Model).

Leider kommt man aber auch nicht umhin, einige gravierende Nachteile von CSS aufzuführen:

- CSS sind eine relativ neue Entwicklung und werden bei weitem nicht von jedem Browser unterstützt. Der Microsoft Internet Explorer ab 3.0 und der Netscape Navigator ab 4.0 beherrschen grundsätzlich den Umgang mit Stylesheets. Neuere Versionen kommen der Originalspezifikation des W3C sogar erstaunlich nah. Aber die volle CSS-Unterstützung bietet noch kein existierender Browser; einige unterstützen CSS gar nicht.

  Die Verwendung von CSS ist also mit Vorsicht zu genießen: Wenn Sie auf gleichzeitige Layouteinstellungen mit alten »HTML-Bordmitteln« verzichten, dann ist in einem CSS-unfähigen Browser gar keine Formatierung mehr zu finden. Im Zweifelsfall müssen also zwei Versionen jeder Seite existieren, was erheblichen Mehraufwand bedeutet.
- Das pixelgenaue Platzieren hat ebenfalls seine Tücken: Da die Monitor- beziehungsweise Browserfenstergröße bei jedem Benutzer anders ist, gibt es keine allgemein optimalen Werte.

16.2.1 Platzieren von Stylesheets

Stylesheet-Angaben können an verschiedenen Stellen im Dokument oder in einer eigenen Datei stehen. Sie haben folgendes Schema:

```
selektor {
   attribut: wert;
   attribut: wert
}
```

Der **Selektor** gibt an, welches Objekt oder welche Gruppe von Objekten durch diese Stylesheet-Angabe formatiert werden soll. Es werden vier verschiedene Arten von Selektoren verwendet:

- **Element**. Hier wird der Name eines HTML-Tags angegeben, und zwar ohne seine Klammern. Beispiele: `h1` oder `td`. Jedes Auftreten des Tags wird automatisch mit den angegebenen Formaten versehen.

  Möglich ist auch eine durch Komma getrennte Liste mehrerer HTML-Tags, für die gemeinsame Einstellungen gelten sollen. Es ist auch kein Problem, differierende Einstellungen noch einmal separat für jedes dieser Tags vorzunehmen.

- **Klasse**. Der Name einer Klasse beginnt mit einem Punkt. Ein HTML-Tag kann mit dem speziellen Attribut `class="klassenname"` versehen werden, damit es zusätzlich zu seinem normalen Stil diese speziellen Einstellungen erhält. Sie können sowohl eine Klasse für ein bestimmtes HTML-Tag einrichten (etwa `h2.blau`), die entsprechend nur für diese Art von Elementen verwendet werden kann, oder aber eine allgemeine Klasse wie `.zitat`, die Sie jedem beliebigen Tag zuweisen können.

  Allgemeine CSS-Klassen werden häufig im Zusammenhang mit zwei besonderen Tags verwendet: `<div> ... </div>` ist ein absatzbildendes Element, genauer gesagt sogar eine frei schwebende, beliebig positionierbare Ebene (»DHTML-Layer«). `<span> ... </span>` kann dagegen beliebige Elemente innerhalb eines Absatzes umfassen und ihnen auf diese Weise Stylesheet-Formate zuweisen.

- **Unabhängiger Stil**. Ein unabhängiger Stil wird einem einzelnen Element im Dokument zugewiesen. Sein Name beginnt mit einer Raute (#); das gewünschte Element erhält diesen Stil über das Universalattribut `id="stilname"` (hier ohne Raute). Am häufigsten werden unabhängige Stile verwendet, um über das Tag `<div>` DHTML-Layer zu erzeugen. In diesem Fall darf eine solche `id`-Definition nur für ein einziges Element im Dokument verwendet werden.

- **Pseudoformate.** Diese Sonderform der Stylesheet-Angabe definiert die verschiedenen Stadien von Hyperlinks oder verschiedene Bereiche von Absätzen. Die Schreibweise für jeden der Hyperlink-Selektoren ist `a:zustand`, wobei der Zustand einen der folgenden Werte annehmen kann:

  - `a:link` ist der Grundzustand des Hyperlinks.
  - `a:visited` beschreibt bereits besuchte Hyperlinks.

- `a:hover` bezeichnet einen Hyperlink, der gerade vom Mauszeiger berührt wird; dieser Zustand ermöglicht einen Rollover-Effekt für Hyperlinks, allerdings nur im Internet Explorer.
- `a:active` ist der aktuell angeklickte Hyperlink.

Der CSS-Code kann an folgenden Stellen definiert werden:

- In einem `<style> ... </style>`-Bereich im Head des Dokuments. Konkret sieht eine solche Angabe folgendermaßen aus:

```
<style type="text/css">
<!--
   /* Konkrete CSS-Angaben */
-->
</style>
```

  Die Verschachtelung der eigentlichen Stylesheet-Angaben in einen HTML-Kommentar ist üblich und verhindert, dass alte, CSS-unfähige Browser diese Angaben im Body der Seite ausgeben. Innerhalb solcher CSS-Blöcke und in externen CSS-Dateien können Sie Kommentare im C-Stil verwenden, wie oben gezeigt:

```
/* Kommentar */
```

- In einer externen Datei, einer Textdatei mit der Dateiendung `.css`. Eine solche Datei darf nur CSS und keinerlei HTML enthalten, auch keine `<style>`-Tags. Eingebunden wird eine solche Datei im Head eines HTML-Dokuments, und zwar folgendermaßen:

```
<link rel="stylesheet" href="format.css" />
```

  Eine moderne Alternative für das Einbinden von externen CSS-Dateien, die allerdings leider nicht von allen Browsern unterstützt wird, ist die Verwendung einer XML-Steueranweisung. Diese kann nicht nur in HTML-Dokumenten, sondern in beliebigen XML-Dokumenten stehen und ein Stylesheet einbinden, das das Aussehen der diversen XML-Elemente definiert:

```
<?xml-stylesheet type="text/css" href="format.css"?>
```

  Die Verwendung externer Stylesheet-Dateien ist besonders vorteilhaft, um einer umfangreichen Website ein einheitliches Layout zuzuweisen, das darüber hinaus an einer zentralen Stelle geändert werden kann.

- Innerhalb eines einzelnen HTML-Tags. Dazu dient das spezielle HTML-Attribut `style="..."`. Beispiel:

```
<p style="...">
<!-- Absatz mit spezieller Formatierung -->
</p>
<p>
<!-- Hier gilt wieder das normale Absatzformat -->
</p>
```

16.2.2 Stylesheet-Wertangaben

Bevor die einzelnen Attribute beschrieben werden, sollten Sie sich einen Überblick über die Arten von Werten verschaffen, die Attribute überhaupt annehmen können. Sie unterscheiden sich zum Teil erheblich von den Werten, die Sie traditionellen HTML-Attributen zuweisen können: Zum einen haben Sie oft viel mehr Auswahlmöglichkeiten, zum anderen sind die Angaben in der Regel konkreter und eindeutiger.

Im Einzelnen sind die folgenden Arten von Werten möglich:

- **Festgelegte Werte**. Viele Attribute kennen eine Reihe unterschiedlicher konkreter Angaben. Beispielsweise erhält das Attribut `font-family` eine Liste bevorzugter Schriftarten. Ein anderes Beispiel ist die Eigenschaft `text-align`, die die Textausrichtung angibt. Sie kann die festgelegten Werte `left`, `right`, `center` und `justify` annehmen.
- **Numerische Werte**. Die Angabe numerischer Werte erfordert eine Zahl und eine angehängte Maßeinheit ohne trennendes Leerzeichen. Tabelle 16.5 zeigt die möglichen Maßeinheiten.

Maßeinheit	Bedeutung	Hinweise
px	Pixel	Die häufigste und wichtigste Maßeinheit
pt	Punkt	Der DTP-Punkt, $^1/_{72}$ Zoll. Wird gern für die Schriftgröße verwendet.
mm	Millimeter	Diese Angaben sind von der Bildschirmgröße und -auflösung abhängig. Gebrochene Werte mit Punkt trennen: `2.5cm`, nicht `2,5`.
cm	Centimeter	
in	Inch (1" = 2,54 cm)	
em	Breite des M	Entspricht dem Geviert, also der Höhe der aktuellen Schriftart.
ex	Breite des x	Wird manchmal als relative Angabe für Wort- oder Zeichenabstände verwendet.
%	Prozent	Im Allgemeinen relativ zum umgebenden Element (Tabellenzelle, Browserfenster). Bei Schriftangaben relativ zur Schriftgröße.

Tabelle 16.5 Maßeinheiten für numerische Stylesheet-Werte

- **Farben**. Zulässig sind die bereits weiter oben beschriebenen HTML-Farbangaben. Eine andere Möglichkeit ist `rgb(rotwert, grünwert, blauwert)`. Die drei Werte können dezimal zwischen 0 und 255 oder in Prozent angegeben werden.

16.2.3 Stylesheet-Eigenschaften

In diesem Unterabschnitt werden die wichtigsten Stylesheet-Attribute aufgelistet und erläutert. Wichtig ist, dass nicht alle Formatangaben bei jedem HTML-Tag zulässig sind. Bei den meisten Attributen ist relativ offensichtlich, für welche Tags sie geeignet sind; bei anderen finden Sie entsprechende Hinweise.

Textformatierung und Typografie

Die folgenden Attribute beschreiben die verschiedenen Aspekte der typografischen Auszeichnung. Sie sind bei fast jedem beliebigen Tag zulässig.

- `font-family`
  Eine durch Komma getrennte Liste von Schriftarten; entspricht der traditionellen HTML-Angabe `<font face="...">`. Beispiel:

  ```
  font-family: Verdana, Arial, Helvetica, sans-serif
  ```

- `font-size`
  Die Angabe der Schriftgröße, meist in Punkt (`12pt`) oder Pixeln (`20px`). Die Pixel-Angabe ist zu bevorzugen. Sie wird plattformübergreifend identisch interpretiert, während Windows und MacOS unterschiedliche Vorstellungen von der Abbildung der Punkt-Angabe auf dem Monitor haben.

Ein Vorteil der klassischen HTML-Größenangaben besteht darin, dass sich die Schrift per Browsereinstellung vergrößern oder verkleinern lässt, während sie relativ zueinander gleich bleiben – das ist besonders für Menschen mit eingeschränkter Sehfähigkeit wichtig. Mit Stylesheets lässt sich dies nachahmen, wenn Sie Schriftgrößen in Prozent angeben. Setzen Sie Fließtext auf `100%` und Überschriften entsprechend größer.

- `font-style`
  Dient dazu, Schrift kursiv zu setzen. Die festgelegten Werte sind folgende: `normal`, `italic` (kursiv), `oblique` (elektronisch schräggestellt, kein separater Kursiv-Schnitt).
- `font-weight`
  Gibt an, wie fett die Schrift dargestellt werden soll. Mögliche Werte sind entweder die festgelegte Angaben `normal`, `bold` (fett) und `extra-bold` (besonders fett) oder glatte numerische Hunderterschritte von 100 (extra light) bis 900 (black, also ultrafett). Die meisten verfügbaren Schriften definieren allerdings keineswegs neun verschiedene Stufen.
- `text-decoration`
  Gibt an, dass der Text mit Linien versehen, das heißt unterstrichen werden soll und Ähnliches: `none` (keine Linie), `underline` (unterstrichen), `overline` (Linie darüber) oder `line-through` (durchgestrichen).
- `letter-spacing`
  Gibt den Zeichenabstand oder die Laufweite des Textes an. Der Standardwert ist `0pt`; positive Werte erzeugen gesperrten Text, negative verengen die Laufweite.

Absatz- und Bereichsformatierung

Die folgenden CSS-Attribute sind zur Formatierung aller absatzbildenden Elemente geeignet, beispielsweise für `<p>`-Elemente, aber auch für Tabellenzellen und ähnliche Tags.

- `text-align`
  Gibt die Textausrichtung mit den möglichen Werten `left` (linksbündig), `right` (rechtsbündig), `center` (zentriert) oder `justify` (Blocksatz) an.
- `text-indent`
  Gibt den Einzug an, also die Einrückung der ersten Zeile eines Bereichs. Der Wert ist numerisch. Positive Werte erzeugen einen eingerückten Einzug, während negative für einen hängenden Einzug sorgen. Für Letzteres ist allerdings eine zusätzliche Verschiebung des linken Innenrandes mittels `margin-left` erforderlich.
- `line-height`
  Bestimmt die Zeilenhöhe durch einen numerischen Wert. Der Standardwert ist etwas größer als die aktuelle Schriftgröße; ein höherer Wert erzeugt einen größeren Zeilenabstand.
- `vertical-align`
  Dies ist die vertikale Ausrichtung des Elements; entspricht dem bekannten `valign` bei Tabellenzellen.
- `display`
  Diese spezielle Angabe ist bei HTML im Grunde überflüssig, da für HTML-Tags klar ist, ob sie absatzbildend sind oder nicht. Wenn Sie CSS dagegen für allgemeine XML-Dokumente einsetzen, ist dieses Format sehr wichtig. Mögliche Werte sind `block`, wenn ein Element absatzbildend sein soll, oder `inline` für ein innerhalb einer Zeile stehendes Element.

Rahmen und Linien

Die folgenden CSS-Attribute regeln die Abstände und Rahmeneinstellungen beliebiger Elemente.

- `margin-top`, `margin-bottom`, `margin-left`, `margin-right`
  Diese vier Attribute bestimmen den Abstand des Bereichs zur Umgebung nach oben, unten, links und rechts. Der Wert ist numerisch. Alternativ gibt `margin` einen identischen Abstand auf allen vier Seiten an.
- `padding-top`, `padding-bottom`, `padding-left`, `padding-right`
  Diese Attribute geben den inneren Abstand eines Elements zu seinem Rand an. Auch hier ermöglicht ein einfaches `padding` eine einheitliche Angabe für alle vier Seiten. Der Unterschied zu den `margin`-Einstellungen zeigt sich, wenn eine sichtbare Rahmenlinie verwendet wird: `padding` setzt den inneren Raum bis zu diesem Rahmen, während `margin` den Außenabstand des Rahmens zur Umgebung festlegt.
- `border-top`, `border-bottom`, `border-left`, `border-right`
  Die Werte für diese Attribute sind eine durch Leerzeichen getrennte Liste von Linienattributen für den äußeren Rand auf der angegebenen Seite eines beliebigen Elements. Stattdessen können Sie mit `border` einen einheitlichen Rahmen auf allen vier Seiten festlegen.

Der erste Wert ist die Linienstärke mit den möglichen Werten `thin` (dünn), `medium` (mittel), `thick` (dick) oder einem beliebigen numerischen Wert. Alternativ kann dieser Wert in Stylesheets auch separat als `border-width` (beziehungsweise `border-top-width` und so weiter) angegeben werden.

Es folgt die Angabe des Linienstils. Möglich sind die folgenden Werte: `solid` (einfache, durchgezogene Linie), `dotted` (gepunktet), `dashed` (gestrichelt), `double` (doppelter Rahmen) sowie die verschiedenen Relief-Effekte `groove`, `ridge`, `inset` und `outset`. Diese Angabe können Sie auch einzeln über das Attribut `border-style` setzen oder Angaben für die Rahmen auf den einzelnen Seiten machen, beispielsweise `border-left-style`.

Schließlich wird die Linienfarbe eingestellt. Hier ist jede beliebige HTML- oder CSS-Farbangabe möglich. Sie lässt sich auch als Einzelwert über `border-color` oder pro Seite durch Attribute wie `border-left-color` festlegen.

Hier zwei Beispiele, um das Ganze zu verdeutlichen:

```
border: medium solid #FF0000
```

Auf allen vier Seiten sollen durchgezogene Randlinien mittlerer Dicke in Rot gesetzt werden.

```
border-top: 4 dashed #0000FF;
border-bottom: 4 dashed #0000FF;
border-left-width: 2;
border-left-style: dotted;
border-left-color: #009900
```

Diese verschiedenen Einstellungen sorgen (unter Nichtberücksichtigung des guten Geschmacks) dafür, dass oben und unten 4 Pixel dicke, gestrichelte blaue Randlinien erscheinen, während sich links eine nur 2 Pixel starke, gepunktete blaue Linie befindet.

Farben und Bilder

Fast jedem Tag, zumindest jedem mit Textinhalt, können mit Hilfe der folgenden Angaben Text- und Hintergrundfarbe oder ein Hintergrundbild zugewiesen werden.

- `color`
  Stellt die Vordergrund- beziehungsweise Textfarbe ein.
- `background-color`
  Definiert die Hintergrundfarbe. Jedes beliebige Element kann indiviuell eingefärbt werden, sowohl absatzbildende Tags als auch Textauszeichnungs-Elemente.
- `background-image`
  Legt ein Hintergrundbild für das Element fest. Der Wert ist die URL des Bildes.
- `background-attachment`
  Bestimmt, ob das Hintergrundbild fixiert wird oder mit der Seite gescrollt wird. Die entsprechenden Werte lauten `fixed` oder `scroll`, wobei Letzteres Standard ist.

- `background-repeat`
  Dieses Attribut bestimmt, ob und in welche Richtung ein Hintergrundbild gekachelt werden soll. Die zulässigen Werte sind `repeat` (in beide Richtungen kacheln), `repeat-x` (nur horizontal wiederholen), `repeat-y` (nur vertikal) oder `no-repeat` (keine Wiederholung, nur Einzelbild).

16.2.4 Layer erzeugen und positionieren

Eine zusätzliche Aufgabe von CSS besteht in der Definition von Eigenschaften für so genannte Dynamic-HTML-Layer. Es handelt sich um unabhängige Ebenen, die frei über der eigentlichen Webseite »schweben« und beliebig positioniert werden können. Nachdem ein Layer erstellt wurde, können seine CSS-Eigenschaften wie Position oder Sichtbarkeit mit Hilfe der Skriptsprache JavaScript nachträglich geändert werden (siehe Kapitel 18).

Technisch gesehen wird ein Layer durch ein `<div>`-Tag gebildet, das dafür ein `id`-Attribut mit einem eindeutigen Namen und die CSS-Eigenschaft `position` benötigt. Praktischerweise kann das `id`-Attribut auch gleich zur Definition der Stylesheet-Formatierung für den Layer über eine unabhängige Stildefinition verwendet werden.

Für Layer werden die folgenden speziellen Formate definiert:

- `position`
  Abgesehen davon, dass ein `<div>` ohne diese Angabe kein Layer, sondern lediglich eine Art Absatz ist, legt das Attribut über seinen Wert die Art der Positionsangaben für den Layer fest: Der Wert `absolute` bedeutet, dass die Positionsangaben in Pixeln erfolgen, während `relative` sie in Prozent des Browserfensters angibt.
- `left`
  Die x-Position des Layers, das heißt sein Abstand vom linken Seitenrand. Der Wert wird je nach `position`-Festlegung in Pixeln (zum Beispiel `100px`) oder in Prozent (beispielsweise `30%`) angegeben.
- `top`
  Diese Eigenschaft gibt entsprechend die y-Position des Layers an, also den Abstand vom oberen Rand.

Alternativ können auch `right` und `bottom` (der Abstand vom rechten beziehungsweise unteren Rand) verwendet werden.

- `z-index`
  Dieses Attribut gibt die »Stapelreihenfolge« von Layers an: Je höher der Wert, desto weiter liegt der Layer im Vordergrund. Falls `z-index` weggelassen wird, werden die Layers des Dokuments einfach in der Reihenfolge gestapelt, in der sie im HTML-Code definiert werden.

Layer-Beispiele

Auf dem Bild `test.jpg`, das sich 100 Pixel vom oberen und 100 Pixel vom linken Rand entfernt befindet, soll der Text »Hallo« erscheinen.

Dazu werden zunächst in einem <style>-Block im Head die folgenden unabhängigen Formate definiert:

```
#bild {
   position: absolute;
   top: 100px;
   left: 100px;
   z-index: 1
}
#text {
   position: absolute;
   top: 110px;
   left: 110px;
   z-index: 2;
   font-size: 18pt;
   color: #FFFF00
}
```

Anschließend werden im Body des HTML-Dokuments die beiden Layer definiert:

```
<div id="bild"><img src="test.jpg" width="200" height="200" alt="Bild
mit Text darauf" /></div>
<div id="text">Hallo</div>
```

Das nächste Beispiel versieht einen Text mit einem »Schlagschatten«; der Text wird einmal in Dunkelgrau und einmal in einer anderen Farbe leicht verschoben übereinander gesetzt.

Als Erstes erfolgt wieder die Stylesheet-Definition im Head:

```
#schrift {
   postion: absolute;
   top: 10px;
   left: 10px;
   z-index: 2;
   color: #FF0000;
   font-family: Arial, Helvetica, sans-serif;
   font-size: 24pt;
   font-weight: bold
}

#schatten {
   postion: absolute;
   top: 14px;
   left: 7px;
   z-index: 1;
   color: #333333;
   font-family: Arial, Helvetica, sans-serif;
   font-size: 24pt;
```

```
    font-weight: bold
}
```

Diese Definitionen können nun folgendermaßen für Layers verwendet werden:

```
<div id="schrift">Wenn wir &uuml;ber den Schatten streiten,<br />
&uuml;bersehen wir das Wesentliche.<br />
-- Aesop</div>
<div id="schatten"> Wenn wir &uuml;ber den Schatten streiten,<br />
&uuml;bersehen wir das Wesentliche.<br />
-- Aesop </div>
```

Abbildung 16.5 zeigt die Darstellung beider Layer-Beispiele im gleichen HTML-Dokument. Dazu wurden lediglich die `z-index`-Werte angepasst, um vier verschiedene »Stufen« zu erhalten.

Abbildung 16.5 Beispiel für frei schwebende DHTML-Layers und CSS-Formatierungen

16.3 Grundgedanken zum Webdesign

Es gibt ein grundlegendes Problem, das den wichtigsten Unterschied zwischen Web- und Printdesign ausmacht: Beim Printdesign wird ein Layout für ein Ausgabemedium einer festgelegten Größe erstellt, sodass die Platzierung von Inhalten und Elementen von vornherein millimetergenau erfolgen kann. Beim Webdesign dagegen ist das Ausgabemedium der Browser des einzelnen Betrachters; Größe, Auflösung und Farbtiefe sowie die unterstützten technischen Fähigkeiten eines solchen Programms können vollkommen verschieden sein.

Index

!, Operator 235
!=, Operator 236
!~, Perl-Operator 294
#, Perl-Kommentar 271
#define, Präprozessor-Direktive 251
#endif, Präprozessor-Direktive 251
#ifdef, Präprozessor-Direktive 251
#include, C-Präprozessor-Direktive 230
#include, Präprozessor-Direktive 250
$, Operator in RegExp 293
$, Perl-Variablen 266, 267
$, PHP-Variablen 818
$/, Perl-Variable 517
$_, Perl-Variable 269
$0, UNIX-Systemvariable 174
%, Operator 235
%, Perl-Variablen 268
&&, Operator
als Perl-Fallentscheidung 273
C 235
Perl 273
&, Dereferenzierungs-Operator 245
&, Operator, C 236
*, Operator 235
RegExp 291
++, Operator 237
+, Operator 235
JavaScript-String-Verkettung 859
RegExp 291
String-Verkettung, Java 255, 256
-, Operator 235, 247
--, Operator 237
., Perl-Operator 272
.htaccess-Dateien, Apache 657
.NET-Framework 48
/, Operator 235
//, Java-Kommentar 257
/etc/exports, NFS-Konfigurationsdatei 648
/etc/passwd, Konfigurationsdatei 172
/etc/shadow, Konfigurationsdatei 173
<a>, HTML-Tag 735
<applet>, HTML-Tag 755
<area>, HTML-Tag 741
<b>, HTML-Tag 727
<body>, HTML-Tag 721

, HTML-Tag 722
<caption>, HTML-Tag 743
<colgroup>, HTML-Tag 747
<dd>, HTML-Tag 733
<dl>, HTML-Tag 733
<dt>, HTML-Tag 733
<embed>, HTML-Tag 753
<font>, HTML-Tag 728
<form>, HTML-Tag 750
<frame>, HTML-Tag 756
<frameset>, HTML-Tag 755
<h1> bis <h6>, HTML-Tags 726
<head>, HTML-Tag 720
<html>, HTML-Tag 720
<i>, HTML-Tag 727
<img>, HTML-Tag 739
<input>, HTML-Tag 751
, HTML-Tag 730
<map>, HTML-Tag 741
<meta>, HTML-Tag 723, 759
<object>, HTML-Tag 754
, HTML-Tag 731
<p>, HTML-Tag 725
<pre>, HTML-Tag 726
<script>, HTML-Tag 856
<select>, HTML-Tag 752
<style>, HTML-Tag 764
<table>, HTML-Tag 742
<td>, HTML-Tag 743
<textarea>, HTML-Tag 752
<th>, HTML-Tag 743
<title>, HTML-Tag 721
<tr>, HTML-Tag 743
, HTML-Tag 730
=, Operator 237
==, Operator 236
=~, Operator 294
=~, Perl-Operator 276
? ,, Operator 238
?, Operator
RegExp 291
@, Perl-Variablen 267
@_, Perl 274
^, Operator
C 236
in RegExp 291, 293
_framesLoaded, ActionScript-Eigenschaft 970
_totalFrames, ActionScript-Eigenschaft 969
``, Operatoren 297
{}, Anweisungsblock 239
|, Operator 236
in RegExp 293
||, Operator 235
als Perl-Fallentscheidung 272
Perl 272
~, Operator 236
0190-Dialer 978
1:1-Relation, RDBMS 359
1:n-Relation, RDBMS 359
10 Base 2, Ethernet-Standard 560
10 Base 5, Ethernet-Standard 561
10 Base T, Ethernet-Standard 562
100 Base T, Ethernet-Standard 562
1000 Base FL, Ethernet-Standard 563
1000 Base TX, Ethernet-Standard 563
16-Bit-Anwendungen, unter Win32 215
3D Now! (CPU-Befehlserweiterung) 105

3-D-Animation 467
Arbeitsablauf 467
Echtzeit-Rendering 467
Film-Rendering 467
Inverse Kinematik 469
Kamera 468
Keyframes 470
NURBS 470
Objekte bewegen 468
Radiosity 474
Raytracing 474
Rendering 468
3-D-Grafik 467
Anwendungsbereiche 467
Beleuchtung 468
in Director 493
Materialien 467
Modeling 467
Texturen 467
3DPI, Director-Xtra 493
3ds max
Animation 473
Ansicht verändern 470
Arbeitsoberfläche 469
Beleuchtung 473
Drehobjekte 471
Kamera 473
Kameraperspektive 473
Keyframes 470
Maps, Material 472
Material zuweisen 472
Modeling 470
Objekte erstellen 469
Objekte modifizieren 469
Objekthierarchie 469
Primitive (geometrische Figuren) 470
Rendering 474
Splines 471
8.3 (MS-DOS-Dateinamensschema) 169

A
Abakus 34
Abfrage, RDBMS
Änderungsabfrage 373
Auswahlabfrage 361, 373
Einfügeabfrage 373
Löschabfrage 373
Abfragen, RDBMS, SQL 362
Abgeleitete Klasse 260
Absatz
Ausrichtung 431
Blocksatz 431
Hurenkind 433
linksbündig 431
rechtsbündig 431
Schusterjunge 433
Typographie 430
Zeilenfall 432
zentriert 431
Absätze
HTML 725
Absoluter Pfad 165, 168
Absorption, Farbe 388
accept(), Perl-Funktion 669
Access Point 991
Access, Datenbank 365
Acrobat Distiller 448
ActionListener 313
actionPerformed(), AWT-Methode 313
ActionScript 952
_framesLoaded, Eigenschaft 970
_totalFrames, Eigenschaft 969
Aktionen-Palette 953
attachMovie (), Methode 968
Color, Objekt 965
createEmptyMovieClip(), Methode 968
Dauerfeuer-Buttons 964
duplicateMovieClip(), Methode 967
Event Handler 954
Event-Handler 954
Farbeffekte 966
Filme laden 956
Formularfunktionen 964
getURL(), Funktion 963
hitTest(), Methode 967
Hyperlinks 963
Instanzen, Eigenschaften 962
Kollisionen ermitteln 967
loadMovie()-Funktion 956
Mauszeiger ausblenden 965
Mouse, Objekt 965
Mouse.hide(), Methode 965
Mouse.show(), Methode 965
Movieclip-Eigenschaften 962
Movieclip-Hierarchie 957
Movieclips automatisch erzeugen 967
Movieclips einfärben 965
Movieclips steuern 957
Movieclips zur Steuerung 965
Navigation 955
on, Anweisung 954
onClipEvent()-Handler 965
stop(), Funktion 953
Zeichenfunktionen 968
Active Directory 646, 991
Active Server Pages → ASP.NET
ActivePerl 265
ActiveX 991
ActiveX-Objekte 754
Ada, Programmiersprache 35
Addierer (Schaltung) 77
Addiermaschinen 34
Addierwerk (Schaltung) 77
Addition, Operator 235
Administrator, Windows-Benutzer 224
Administratorendokumentation 336
Admin-Rechte 976
Adobe
Acrobat 447
Audition 455
Illustrator 387, 412
InDesign 437, 440
PDF-Format 447
Photoshop 387
Photoshop Elements 387
PostScript 526
Type Manager (ATM) 440
Adobe PostScript 141
Adressbus 101
Wortbreite 102
Adressierung, Speicher 69
ADSL 573

AdWare 979
afpd, netatalk-Dienst 651
Aggregatfunktionen, SQL 379
Agile Softwareentwicklung 338
AGP 119, 991
AI → Künstliche Intelligenz
AIFF, Audio-Dateiformat 533
AIX, Betriebssystem 148
Akteure (UML) 341
Aktivitätsdiagramme (UML) 344
Akustikkoppler 541
Al Chwarismi (arab. Mathematiker) 33
Algebra
boolesche 55, 80
Definition 80
lineare 80
zur Algorithmendarstellung 80
Algorithmen 33, 80, 277
algebraische Darstellung 80
Anschaulich-sprachliche Darstellung 81
Berechenbarkeit 81
Binäre Suche 285
BubbleSort 279
Diagrammdarstellung 81
entwickeln 277
Komplexität 82
Lineare Suche 83, 284
O-Notation der Komplexität 83
Permutationen 83
Pseudocode-Darstellung 81
QuickSort 282
Sortieren 279
Suche 284
Algorithmus 991
Größter gemeinsamer Teiler (GGT) 277
Alias (MacVerknüpfung) 211
ALOHANet 559
Alpha, Prozessor 103, 105
Alphabet
Geschichte 421
Griechisches 422
Lateinisches 422
Phönizisches 421
Alphakanal 991
Photoshop 402
ALRM, Signal 159
Altair 8800, früher Mikrocomputer 39
ALU 100, 991
AMD 100
Amiga 41
Amigos, drei 339
Amplitude, Audio 50
Analog, Unterschied zu Digital 48
Analoguhr, JavaScript-Anwendung 909
Analyse, Software-Engineering 330
Lastenheft 331
Objektorientierte Analyse 331
Pflichtenheft 332
Strukturierte Analyse 331
Analytical Engine 34
Änderungsabfrage 373
AND-Schaltung 75
Aufbau mit Transistoren 75
mit einfachen Mitteln nachbauen 73
AND-Verknüpfung 56
Wahrheitstabelle 57
Anführungszeichen
PHP 820
verschiedene, Bedeutung in Perl 271
Angewandte Informatik 26
Animation
Doble Buffering 309
Fahrrad, Flash 939
Flash 928
Java, AWT 309
Tweening, Flash 932
Animierte GIFs 785
Anschnitt, Bilder im 436
ANSI 991
C-Standard 248
ANSI-Zeichensatz 520
Anti-Alias 991
Photoshop 400
Antiqua 426, 991
Antiqua-Varianten 428
Antivirenprogramme 974
Antville, Blog-Tool 809
Anweisungen, C 231
Anweisungsblock 239
Anwenderdokumentation 336
Anwendung
DDN-Modell-Schicht 547
OSI-Schicht 545
Anwendungsfalldiagramme (UML) 341
Anwendungsfälle 336
Anwendungsserver 557
Apache
.htaccess-Dateien 657
Authentifizierung 663
Autostart (UNIX) 202
Direktiven 658
Filter 653
Grundkonfiguration 663
htpasswd-Dienstprogramm 659
HTTP-Server 652
Installation, UNIX 655
Installation, Windows 656
Konfiguration 657
Module 653
Multi-Processing-Module 653
Multiprotokoll-Support 653
Verzeichnisschutz 663
Virtuelle Hosts 662
Xalan 710
XAMPP, integrierter Server 652
Xerces 712
Apache Portable Runtime (APR) 653
API 991
Apple
MacOS X 206
Macintosh 150
QuickTime 207
Apple II 40, 149
Apple Macintosh 41

Apple-Menü, Wichtige Befehle 209
AppleTalk 629, 991
Einrichten, Mac OS X 638
Protokollstapel 630
Vergleich zu TCP/IP 631
Applets, Java 251, 304, 316
Application Server 557, 991
Java 675
APR → Apache Portable Runtime (APR)
Aqua, Mac OS X-Oberfläche 148, 207
Arabische Zahlen 34
Arbeitsspeicher 98
des virtuellen Prozessors 88
Arbeitsverzeichnis 165
anzeigen, UNIX 181
wechseln, UNIX 181
wechseln, Windows 222
Archivdateien, tar 205
ARCNet
Netzwerk 569
A-Record (DNS) 613
ArgoUML, Tool 340
Arithmetic-Logical Unit → ALU
Arithmetische Operatoren
C 235
SQL 378
Arithmetisch-logische Einheit → ALU
ARP 991
TCP/IP-Netzzugang 577
ARPA 539, 991
ARPANet 539
Anwendungen 540
MilNet 541
technische Grundidee 539
ursprüngliche Aufgabe 539
Arrays 245, 992
Anzahl der Elemente, Perl 280
aus Strings, Perl 269
C 245
Deklaration, C 245
mehrdimensionale, PHP 821
Perl 267
PHP 819
zur C-String-Darstellung 246
zusammenfassen, Perl 269
Artificial Intelligence → Künstliche Intelligenz
AS, SQL-Klausel 378
ASCII 992
Zeichensatz 51
ASCII-Art 727
ASCII-Code 518
Erweiterungen 519
IBM-Erweiterung 519
Steuerzeichen 519
Tabelle 519
ASCII-Modus 992
ASP.NET 808, 992
Assembler 43, 992
Mnemonics 44
Nicht-x86 107
Praktische Anwendung 44
x86-Beispiele 107
Asymmetrische Verschlüsselung 987
AT&T 46, 147
UNIX System V 148
AT&T Bell Laboratories 46
atalkd, netatalk-Dienst 651
Atari 41, 149
Atari 800XL 41, 149
Atari ST 41
AT-Befehlssatz 992
Athlon, Prozessor 100, 105
ATM (Adobe Type Manager) 440, 992
ATM, Asynchronous Transfer Mode 568
atomar, Information in RDBMS 362
attachMovie (), ActionScript-Methode 968
attrib, Windows-Befehl 222
Attribute (Windows-Dateien) 168
Attribute, HTML 720
Audio
Adobe Audition 455
Arrangements 458
Aufnahme 454
Aufnahmequelle wählen 454
Digitale Bearbeitung 453
Digitalisierung 453
in Director 493
in Flash 943
MIDI-Sequencer 454
MIDI-Tracks einfügen 455
Wave-Editoren 454
zeit- und wertdiskret 51
Audio-CD 129
über Soundkarte abspielen 142
Audio-Dateiformate 533
AIFF 533
MP3 534
MP4 534
OGG Vorbis 534
WAV 533
Audiodaten 50
Sampling 50
Sampling-Rate 51
Sampling-Tiefe 51
Tonkanäle 51
Audition, Wave-Editor 455
Amplitudeneffekte 457
Effekte 456
Sound-Arrangements 458
Störungen entfernen 457
Tracks bearbeiten 455
Wellen- und Spektralansicht 456
Zeit- und Tonhöhen-Effekte 457
Auflicht-Scanner 135
Auflösung
bei der Digitalkamera 136
der Grafikkarte 137
Auflösung, Bild 50
Aufnahme, Audio 454
Ausbildung 26
Fachinformatiker 27
Informatikkaufleute 28
IT-Systemelektroniker 28
IT-Systemkaufleute 28
Mediengestalter 29
Prüfung 28
Studienfächer 31
Ausdruck 992

Ausdrücke
Bedingungen 239
C 234
Perl 270
Ausgabe
in Dateien, C 249
Konsole, C 248
Perl 266
Ausgabeeinheit 97
Ausgabegeräte 123, 136
Drucker 140
Grafikkarte 137
Monitor 138
Ausgabesteuerung, durch das Betriebssystem 145
Ausgabeumleitung
in UNIX-Shells 177
Windows 221
Auslagerungsdatei 162
Ausnahmen
auslösen 263
FileNotFoundException, Java 264
IOException, Java 264
Java 255, 263
Java, IOException 255
Ausrichten, Illustrator 418
Aussage 992
Aussageformen 54
Aussagelogik 53
Aussagen 53
falsche 54
mathematische 54
wahre 54
Auswahl, Illustrator 413
Auswahl, Photoshop 399
Ellipse 399
Glätten 400
Lasso 400
Maskierungsmodus 403
mit Kanal 402
Rechteck 399
Weiche Kante 400
Zauberstab 400
Auswahlabfrage 361, 373
Authentifizierung, Apache 663
AUTO_INCREMENT, SQL-Feldoption 377
Autoflush-Modus 668
Automatentheorien 25, 79
Registermaschine 87
Turing-Maschine 85
Automatisch starten
System V Init 202
UNIX 202
Automatische Variablen 234
Autonomes System 992
Autonomes System (AS), Routing 592
Autorensysteme 474
Average Case (Komplexität) 82
AVI, Video-Dateiformat 535
awk 264, 290
AWT 992
AWT, Java 304
Ereignisbehandlung 304

B

Babbage, Charles 34
Back Orifice, Backdoor 977
Backdoors 977
background-attachment, CSS-Angabe 768
background-color, CSS-Angabe 768
background-image, CSS-Angabe 768
background-repeat, CSS-Angabe 769
Backtick, Operator 297
Backups 975
Band, der Turing-Maschine 85
Barock-Antiqua 426
bash (Borne Again Shell) 174
BASIC 45
Basic Input/Output System → BIOS
BASIC, Programmiersprache 39, 45
Basis, bei Stellenwertsystemen 65
Bastion-Host 985
Batch Processing → Stapelverarbeitung
Batch-Datei, Windows 223
Baum, Datenstruktur 288
Binärbaum 288
Baum-Topologie, Netzwerk 552
BCD → Binaray Coded Decimals
BEA WebLogic-Server 675
Beck, Kent 338
Bedingter Sprung 106
Befehle
Dateiverwaltung, UNIX 179
des virtuellen Prozessors 89
Systemverwaltung, UNIX 185
Textmanipulation, UNIX 183
UNIX 179
Windows-Konsole 222
Befehlstabelle 101
der CPU 101
Befehlszeiger (CPU-Register) 100
Behavioral Patterns → Verhaltensmuster
Bell Laboratories 147
Bemer, Robert 518
Benutzer
Administrator, Windows 224
an- und abmelden, Windows 219
Gruppe hinzufügen, UNIX 188
hinzufügen, UNIX 188
Home-Verzeichnis 165, 168
Passwort ändern, UNIX 188
root 159, 172
Verwaltung, MacOS X 212
Verwaltung, Windows 224
Zugriffsrechte 166
Benutzermodus 153, 158
Benutzeroberfläche 145
grafische 146, 150, 157
Konsole 146, 157
Benutzerrechte, ändern, UNIX 182
Berechenbarkeit 25, 992
Berechenbarkeit von Algorithmen 81

Berechenbarkeit, Halteproblem 81
Berkeley Socket API 664, 992
Berkeley, Universität 148, 193, 664
UNIX-Version 148
Berners-Lee, Tim 541
Beschaffungsmanagement 329
Beschreibbare DVDs 133
Besitzer, wechseln (Datei) 183
Best Case (Komplexität) 82
Betriebssystem
Aufbau 151
Aufgaben 145
Benutzermodus 153
Benutzeroberfläche 145
Bibliothek 156
Booten 153
BSD-UNIX 148
CP/M 149
Darwin 148, 207
Dateiverwaltung 145
Dialogverarbeitung 147
Ein- und Ausgabesteuerung 145
FreeBSD 148
Gerätetreiber 154
Geschichte 146
herunterfahren 190
HP UX 148
IBM AIX 148
ITS (Incompatible Timesharing System) 147
Kernel 152
Kernelmodus 153
Konsole 157
Linux 148, 149, 151, 169
Mac OS X 148, 206
Minix 151
MS-DOS 150
MULTICS 147
Multitasking 154, 158
Neustart 190
OS/2 150
Prozessmanagement 145
Shell 157
Speichermanagement 145
Stapelverarbeitung 147
Sun Solaris 148
Systemaufrufe 108, 153, 155
Systemprogramme 155
Task Scheduler 153
Threads 152, 161
Timesharing 147
Unicode-Unterstützung 522
UNIX 46, 148
UNIX System V 148
Verwaltungsbefehle, UNIX 185
VMS 150
Win32 API 156
Windows 213
Windows 2000 151, 214
Windows 95 151, 214
Windows 98 151, 214
Windows Me 151, 214
Windows NT 150, 214
Windows Server 214
Windows XP 151, 214
Windows, Versionsübersicht 214
Zeichensatzeinstellung 523
Bewegungsdaten 355
Bezeichner 992
C 232
in PHP 818
Bézier, Pierre 412
Bézierkurven → Pfade
BGP 992
BGP, Routing-Protokoll 594
Bibliothek des Betriebssystems 156
Big-Endian-Architektur 992
Big-Endian-Plattformen 527
BIGINT, SQL-Datentyp 375
Bikubische Interpolation, Bildgröße 396
Bildauflösung 392, 992
Druckauflösung 392
im Webdesign 778
Photoshop 395
Scannen 394
Bildbearbeitung 387
Alphakanal 402
Anti-Alias 400
Arbeitsflächengröße 397
Auswahl 399
Bildauflösung 392
Bildauflösung einstellen 395
Bildgröße 392
Bildgröße einstellen 395
Bildgrößeninterpolation 396
Bildschirmauflösung 393
Farbkorrektur 403
Farbmodelle 389
Farbmodus 398
Farbtiefe 391
für Websites 776
Histogramm 404
Praxis 394
Scan-Auflösung 394
Transformation 403
Bilddateiformate 530
BMP 533
GIF 531, 779
im Webdesign 779
JPEG 532, 779
Photoshop 530
PICT 533
PNG 532, 780
PSD 530
TIFF 531
Bilddaten
Auflösung 50
digitale 50
Farbkanal 50
Farbtiefe 50
Bilddatenbank 356
Bilder in HTML einbetten 739
Bildgröße 392
Bikubische Interpolation 396
Bilineare Interpolation 396
im Webdesign 778
Interpolation 396
Photoshop 395
Pixelwiederholung, Interpolation 396
Bildplatzhalter 437
Bildschirmauflösung 393
Bildwiederholrate (Monitor) 138
Bilineare Interpolation, Bildgröße 396
bin, UNIX-Verzeichnis 164

Binärbaum, Datenstruktur 288
C 288
Binärdaten 49, 64
Binäre Suche 285, 992
Java 285
Binärmodus 992
Binary Coded Decimals 73, 992
BINARY, SQL-Feldoption 376
bind(), Perl-Funktion 669
BIND, DNS-Serversoftware 608
BIND-Nameserver 610
A-Record 613
CNAME-Record 614
Installation 610
Konfiguration 611
MX-Record 615
NS-Record 614
PTR-Record 613
Reverse-Lookup-Zone 611
SOA-Record 613
Zonendaten-Dateien 612
Zonendefinition 611
Bio-Computer 42
Bioinformatik 32
BIOS 99, 110, 992
Aufgaben 111
beim Macintosh 114
Betriebssystem starten 111
Boot-Reihenfolge 112
CMOS löschen 112
CMOS-RAM 111
eingebaute Routinen 111
POST 111
POST beschleunigen 112
Power Management einstellen 113
Setup 111
Update 113
BIOS-Setup 111
Einstellmöglichkeiten 112
Einstellungen zurücksetzen 113
BIOS-Update 113
Anleitung 114
Motivation 113
Bit 64
Bit-Komplement, Operator 236
Bitmap 992
Photoshop-Farbmodus 398
Bitmap-Grafik 50, 387
Bitmap-Schrift 440
Bit-Operatoren 236
Bitübertragung, OSI-Schicht 544
Bit-Verschiebung, links 236
Bit-Verschiebung, rechts 236
Bitweise Exklusiv Oder, Operator 236
Bitweise Oder, Operator 236
Bitweise Und, Operator 236
Blindtext 437
in InDesign generieren 438
BLOB, SQL-Datentyp 376
Block, Anweisungen 239
Blockgeräte (block devices) 154
Blocksatz 431
in HTML 726
Silbentrennung 432
Blogs → Weblogs
Blue Book (Mixed-Mode-CD) 130, 993
Bluescreen, Video-Transparenz 465
Bluetooth 123
BMP, Bilddateiformat 533
Booch, Grady 339
Boole, George 55
boolean, Java-Datentyp 256
Boolesche Algebra 55, 80, 993
Java-Datentyp 256
Booten 153
BSD-Startskripte 204
LILO 202
MacOS X-Startvolume 211
System V Init 202
Boot-Reihenfolge einstellen 112
Bootsektor-Viren 973
border, CSS-Angabe 767
BorderLayout 312
Borland 317
Bourne Again Shell (bash) 174
Bourne-Shell 174
Boyce-Codd-Normalform, RDBMS 363
Branch Prediction → Sprungvorhersage
break
C-Anweisung 241
Schleifen abbrechen 283
Bridge 562, 993
Broadcast, IP-Protokoll 579
Browserweiche, DHTML-Stil 912
Brute-Force-Attacke 173, 993
BSD 993
BSD-Startskripte 204
BSD-UNIX 148
FreeBSD 148
BubbleSort, Algorithmus 279
Java 280
Buchstaben
Bestandteile 424
Geschichte 421
Großbuchstaben 424
Kapitälchen 425
Kleinbuchstaben 425
Buffer im Emacs-Editor 196
BufferedReader
Java-Klasse 254, 263
BufferedReader, Java-Klasse, readLine(), Methode 255
Bunte Bücher (CD-Standards) 129
BURN-Proof-Technologie 131
Bus 115
Bluetooth 123
Definition 99
der CPU 101
drahtloser 122
EIDE 120
FireWire 122
Funk-Schnittstelle 123
Hot-Plugging 122
Infrarot 123
IrDA 123
Kartensteckplätze 119
Laufwerksanschlüsse 120
paralleler 122
PS/2 122

RS-232 122
SCSI 121
serieller 122
USB 122
Bus Mastering 118, 993
Bus-Topologie, Netzwerk 552
Button, AWT-Klasse 311
Byron, Ada → Lovelace, Ada
Byte 70
byte, Java-Datentyp 256

C
C 993
C# 48, 993
C++ 48, 993
C, Programmiersprache 46, 148, 228, 235, 243
#define, Präprozessor-Direktive 251
#endif, Präprozessor-Direktive 251
#ifdef, Präprozessor-Direktive 251
#include, Präprozessor-Direktive 230, 250
-, Operator 247
ANSI-Standard 248
Anweisungen 230, 231
Anweisungsblock 239
Arithmetische Operatoren 235
Arrays 245
Ausdrücke 234
Bezeichner 232
Binärbäume 288
Bit-Operatoren 236
break-Anweisung 241
char 233
Compiler 228
Datentypen 233
Datentypkonvertierung 235
Datum und Uhrzeit 250
difftime()-Funktion 250
double 233
Escape-Sequenzen 231
exec()-Funktion 297
Exponentialschreibweise 234
Fallentscheidungen 239
fgets()-Funktion 249
Fließkomma-Datentypen 233
Fließkomma-Literale 234
float 233
Flusskontrolle 236
fopen()-Funktion 249
for()-Schleife 243
fork()-Funktion 296
fprintf()-Funktion 249
fscanf()-Funktion 249
Funktionen 243
Funktionsaufrufe 231, 244
Funktionsrückgabewert 231
ganzzahlige Datentypen 233
Geschichte 228
getchar() 248
gets()-Funktion 231, 248
globale Variablen 234
Header-Datei 248
Hexadezimalzahlen 234
int 233
int, Funktionsdatentyp 230
Integer-Literale 234
Kommandozeilenparameter 244
Kommentare 232
kompilieren 229
Kontrollstrukturen 232, 239
Listen 286
Literale 234
localtime()-Funktion 250
Logische Operatoren 235
lokale Variablen 234
long 233
main()-Funktion 230, 243
malloc()-Funktion 287
mem.h 287
NULL 250
Oktalzahlen 234
Präprozessor 250
printf()-Funktion 230, 248
return-Anweisung 231, 243
scanf()-Funktion 241, 248
Schleifen 242
short 233
Speicher reservieren 287
Standardbibliothek 156, 247
static-Deklaration 234
statische Variablen 234
stddef.h 234
stdio.h 230, 248
strcat(), Funktion 249
strcmp()-Funktion 249
strcpy()-Funktion 249
strftime()-Funktion 250
string.h 249
String-Literale 235
struct 247
Strukturen 247
switch/case 241
Syntax 231
sys/types.h 296
time()-Funktion 250
time.h 250
time_t, Datentyp 250
Umgang mit Whitespace 232
Variablen 233
Variablen, Gültigkeitsbereich 234
Variablendeklaration 230, 232, 233
Vergleichsoperatoren 236
void, Funktionsdatentyp 243
Vorzeichen in Datentypen 233
wchar_t 234
Wertzuweisung 232
while()-Schleife 242
Zeichen-Literale 235
Zeiger 244
C64 41, 149
Cache 101
bei Festplatten 128
Level 1 101
Level 2 101
Call by Reference 245
Call by Value 245
Callback-Methode 302, 993
Canvas, AWT-Klasse 304
CAPTCHAs 978
Carbon 993
MacOS X-API 207
Cäsar-Code 986
Cascading Style Sheets → CSS

case, Befehl in Shell-Skripten 191
CASE-Tools 340
cat, UNIX-Befehl 183
catch(), Java 255
CAV → Konstante Winkelgeschwindigkeit
CBT (Computer Based Training) 452
CCD
bei der Digitalkamera 136
beim Scanner 135
C-Compiler 228
GCC 228
CD
beschreibbare 130
Brennsoftware 131
BURN-Proof-Technologie 131
Datenformate 131
Disc-at-once 130
Hybrid-CD 132
ISO-9660-Format 131
Joliet-Format 131
Lead-In-Area 130
Lead-Out-Area 130
Multisession 130
Track-at-once 130
cd, UNIX-Befehl 181
cd, Windows-Befehl 222
CD-Brennsoftware 131
CDE, Window-Manager 198
CDi 130
CD-R 130
CD-ROM 129
Geschwindigkeit 130
CD-RW 130
CD-Standards 129
Blue Book 130
Green Book 130
Orange Book 130
Red Book 129
White Book 130
Yellow Book 129
CD-Text 129
Central Processing Unit → Prozessor
Centronics-Anschluss 122
CGI 807, 993
Sicherheitsprobleme 984
char, C-Datentyp 233
CHAR, SQL-Datentyp 376
charAt(), Java-Methode 257
Chatbot Eliza 84
chgrp, UNIX-Befehl 183
Child-Prozess 159, 993
Chip 36
Chipsatz 99
chmod, UNIX-Befehl 182
chomp, Perl-Funktion 266
Chomsky, Noam 48
chown, UNIX-Befehl 183
Chroma-Keying, Videotransparenz 465
chroot-Umgebung 985
CHS (Festplattenadressierung) 126
CIDR, IP-Adressierung 581
CIDR-Adressierung 993
circuit switching 538
CISC-Prozessor 104
Beispiele 105
class, Java-Schlüsselwort 254
CLASSPATH, Umgebungsvariable 252
Clear-CMOS-Jumper 112
Client, Netzwerk 554
close, Perl-Anweisung 275
closedir, Perl-Anweisung 275
Cluster (Dateisystem) 163
CLV → Konstante lineare Geschwindigkeit
Cmd.exe, WinNT-Shell 221
CMOS-RAM 111
löschen 112
CMS → Content-Management
CMYK, Photoshop-Farbmodus 399
CMYK-Farbe 50, 389
Farbseparation 390
CNAME-Record (DNS) 614
Cobol 45
Cobol, Programmiersprache 45
Cocoa 993
Cocoa, MacOS X-API 207
Code-Review 335
Color, ActionScript-Objekt 965
Color, AWT-Klasse 307
color, CSS-Angabe 768
Color-Management 388
COM, Microsoft-Komponentenmodell 675
COMMAND.COM, MS-DOS-Shell 221
Commit (Transaktionen) 364
Commodore 41, 149
Commodore Amiga 41
Common Gateway Interface → CGI
Compact Disc 129
compareTo(), Java-Methode 256
Compiler 44, 107
Java 253
Computer Science → Informatik
Computer, Definition 32
Computergrafik 387
Bitmaps 387
Farbmodelle 389
Farbtiefe 391
Programme 387
Vektorgrafik 387
Computersystem, schematischer Aufbau 95
Computerviren → Viren
connect(), Perl-Funktion 668
Connection, JDBC-Klasse 383
Constant Angular Velocity → Konstante Winkelgeschwindigkeit
Constant Linear Velocity → Konstante lineare Geschwindigkeit
Container-Tags, HTML 720
ContentHandler, SAX-Interface 714
Content-Management 808
TYPO3 808
Cookies 993
Cookies, in PHP 834
Coprozessor → Koprozessor

copy, Windows-Befehl 222
CORBA, Komponentenmodell 675
Corel Draw 387
Corporate Design beim Webdesign berücksichtigen 774
Cosinuskurve, zeichnen, AWT 307
COUNT, SQL-Funktion 379
cp, UNIX-Befehl 180
CP/M, Betriebssystem 149
C-Programmiersprache, char, Datentyp 230
CPU 98
alte Bedeutung 99
CPU → Prozessor
CR, Mac-Zeilenumbruch 516
crack (Passwort-Knackprogramm) 172
Crackerangriffe 983
Crackertools 985
CREATE DATABASE, SQL-Befehl 373
CREATE TABLE, SQL-Befehl 373
createEmptyMovieClip(), ActionScript-Methode 968
CreateProcess(), Windows-Systemaufruf 160
Creational Patterns → Erzeugungsmuster
Creator ID, HFS 213
CRLF, Windows-Zeilenumbruch 516
Crosslink-Ethernet-Kabel 562
Cross-Site-Scripting 983
CRT → Röhrenmonitor
csh (C-Shell) 174
C-Shell 174
CSMA/CA 993
CSMA/CA, Netzzugangsverfahren 566
CSMA/CD 993
CSMA/CD, Netzzugangsverfahren 559
CSS 762, 993
Absatzformatierung 767
Abstand vom linken Rand 769
Abstand vom oberen Rand 769
Anzeigeart 767
Aufgabe 762
Ausrichtung 767
Außenrand 767
background-attachment 768
background-color 768
background-image 768
background-repeat 769
Bilder 768
border 767
color 768
display 767
Einzug 767
Element (Tag) formatieren 763
Externe Dateien 764
Farbangaben 766
Farben 768
Feste Werte 765
fett 766
font-family 766
font-size 766
font-style 766
font-weight 766
Formate dynamisch ändern, DOM 902
für XML-Dokumente 709
Hintegrundfarbe 768
Hintergrund befestigen 768
Hintergrund kacheln 769
Hintergrundbild 768
in Dreamweaver 801
Innenabstand 767
Klassen 763
kursiv 766
Laufweite 766
Layer 769
Layer, Beispiele 769
left 769
letter-spacing 766
line-height 767
Linien 767
margin 767
Nachteile 762
Numerische Werte 765
padding 767
position 769
Positionsart 769
Pseudoformate 763
Rahmen 767
Schriftart 766
Schriftgröße 766
Selektor 763
Stapelreihenfolge 769
Struktur 763
style, HTML-Attribut 764
style, HTML-Tag 764
text-align 767
text-decoration 766
Textfarbe 768
Textformatierung 766
text-indent 767
top 769
Unabhängige Stile 763
unterstrichen 766
vertical-align 767
vertikale Ausrichtung 767
Wertangaben 765
Zeilenhöhe 767
z-index 769
CSS Vorteile 762
C-Standardbibliothek 156, 247, 993
CUPS 993
CUPS, UNIX-Drucksystem 648
Scheduler 649
Starten 649
Cutler, David 150
CygWin 228

D

Daemon 189, 993
DARPA 539
Darstellung, OSI-Schicht 545
Darwin, Betriebssystem 148, 207
Data Fork, HFS 212
Datagramm 994
Datagramm-Sockets 665
date(), PHP-Funktion 839
Date, JavaScript-Klasse 873
DATE, SQL-Datentyp 375

date, UNIX-Befehl 187
Formatierung 187
Datei 145
Besitzer wechseln, UNIX 183
Gruppe wechseln, UNIX 183
String lesen aus, C 249
Zugriffsrechte ändern 182
Dateiattribute 168
Windows, ändern 222
Dateien
Attribute ändern 222
Ausgabe in, C 249
bearbeiten, Perl 274
Eingabe aus, C 249
Kopieren 211
kopieren 180, 219, 222
löschen 222
löschen unter UNIX 181
Öffnen, C 249
Schließen, C 249
Umbenennen 211
umbenennen 180, 222
verarbeiten, Java 263
Verschieben 211
verschieben 180, 219, 222
Verwaltung 163
Dateierweiterung 169
Anzeige einschalten 169
Dateiformate 515
AIFF 533
Audio 533
AVI 535
Bilder 530
binäre 527
BMP 533
GIF 531, 779
JPEG 532, 779
MP3 534
MP4 534
MPEG 535
OGG Vorbis 534
PICT 533
PNG 532, 780
PostScript 526
QuickTime 535
Text 515
textbasierte 524
TIFF 531
Video 534
WAV 533
Dateihandles, Perl 275
Dateinamen
Endung 180
Endung sichtbar machen 169
Erweiterung 169, 180
Groß- und Kleinschreibung 165
MS-DOS 169
UNIX-Platzhalter 179
unter UNIX 165, 179
unter Windows 168
Dateiserver 554
Dateisystem, virtuelles 163
Dateisysteme 163
Benutzerrechte 166
CD 131
erzeugen 186
ext3 186
FAT 12 215
FAT 16 215
FAT 32 216
Fehlerprüfung (UNIX) 186
inodes 166
Journaling-Funktion 186
Laufwerksbuchstaben 167
Links 166
Linux 164
MacOS X 212
Mounten 166
mounten 185
NTFS 216
reiserfs 186
UNIX 164
UNIX-Pfadangaben 165
UNIX-Verzeichnisbaum 164
Windows 167, 215
Windows-Pfadangaben 167
Zuordnungseinheiten 163
Dateiverwaltung 145, 163
unter UNIX 179
Dateiviren 973
Daten
Bewegungsdaten 355
Ordnungsdaten 355
Rechendaten 355
Stammdaten 355
Datenanalyse 331
Datenbank 355
1:1-Relation 359
1:n-Relation 359
Abfragen, objektorientierte 367
Access 365
atomare Information 362
Auswahlabfrage 361
Bild- und Multimedia- 356
Boyce-Codd-Normalform 363
Datenarten 355
Datenfeld 357
Datensatz 357
Einzeltabelle 356
Einzeltabelle, Definition 357
Einzeltabelle, Grenzen 358
Entity 357
erzeugen, SQL 373
FileMaker 365
Filterung 358
freie Server 365
Fremdschlüssel 359
Funktionen 358
Grenzen der RDBMS 365
IBM DB2 365
Index 359
Join-Abhängigkeiten 364
kommerzielle Server 365
Konsistenz 358
löschen, SQL 374
m:n-Relation 359
Microsoft SQL Server 365
MySQL 368
Normalformen 362
Normalisierung 362
objektorientierte 356, 365
ODBC 382
ODL 366
OQL 367
Oracle 365
PostgreSQL 365
Primärschlüssel 359
Programmierung 382
relationale 356, 358
Relationen 359
Schlüssel 359

Sortieren 358
SQL 362
Suchen 358
Tabelle erzeugen, SQL 373
Tabelle löschen, SQL 374
Transaktionen 364
Typen 356
Volltextdatenbank 356
XML 356
Datenbanken, SQL 372
Datenbus 101
Wortbreite 102
Datenfeld 357
Datenformate 515
Text 515
Datenkomprimierung 529
Datenpakete 538
Frames 559
Datensatz 357
Datensicherung 975
Datenstrukturen 80, 277, 285
Bäume 288
Queue 285
Stack 285
Datenträgerverwaltung, Windows 225
Datentypen
boolean, Java 256
ganzzahlige, C 233
Java 255
Umwandlung, C 235
Variablen, C 233
Datenübertragung
Paketvermittlung 538
parallele 115
Schaltkreisvermittlung 538
serielle 115
Datenübertragung, Geschwindigkeitsmessung 117
Datenverarbeitung
Elektrische 25
Elektronische 25
Manuelle 25
Mechanische 25
DATETIME, SQL-Datentyp 375
Datum und Uhrzeit
C 250
date-Befehl, UNIX 187
Differenzen berechnen 250
EPOCH 156
Ermitteln, C 250
formatieren (UNIX) 187
Formatieren, C 250
JavaScript 873
PHP 839
SQL 375
DB2, RDBMS 365
DBMS (Database Management System) 355
dd (Didot-Punkt) 422
DDN-Schichtenmodell 545
Anwendungsschicht 547
Host-zu-Host-Transportschicht 546
Internetschicht 546
Netzzugangsschicht 546
DDR-RAM 109
De Marco, Tom 331
Deadlock 160, 994
Debian GNU/Linux 171
DEC 46
DEC → Digital Equipment Corporation (DEC)
DECIMAL, SQL-Datentyp 375
Deckkraft, Photoshop 406
default
switch/case-Vorgabewert 241
Default Gateway 994
DEFAULT, SQL-Feldoption 377
defined, Perl 273
Defragmentierung 994
Defragmentierung (Festplatten) 126
Deklaration 994
Deklarative Programmiersprachen 48
del, Windows-Befehl 222
DELETE, SQL-Abfrage 381
DeMorgan-Theorem 58, 994
Denial of Service 983
Denial of Service, Netzwerkangriff 642
Denial of Service-Angriff 994
Deployment 994
Dereferenzierung, Perl 270
Design Patterns → Entwurfsmuster
Desktop Publishing 421
dev, UNIX-Verzeichnis 164
Dezimalsystem 65
in Dual umrechnen 67
in Hexadezimal umrechnen 68
DFÜ
Akustikkoppler 541
Mailboxen 541
Online-Dienste 541
Praxis 569
über ISDN 573
DFÜ, PPP-Protokoll 570
DHCP 595, 994
DHCP-Server, Windows Server 2003 646
DHTML 896, 994
Browserübergreifende Lösungen 909
in Dreamweaver erstellen 802
Layer-Inhalte austauschen 912
DHTML-Layer, manipulieren, DOM 903
Diagrammtypen (UML) 339
Dialer 978
Dialogverarbeitung 147
Diascanner 136
Diashow, JavaScript-Anwendung 883
Dickte 994
Dickte, Typographie 423
Didot, François 422
Didot-Punkt, Schriftgröße 422
Dienste, Windows 225
diff, UNIX-Befehl 185
Differential Engine 34
difftime(), C-Funktion 250
dig, TCP/IP-Dienstprogramme 643
Digital Equipment Corporation (DEC) 38
Digital Versatile Disc → DVD
Digital, Unterschied zu Analog 48

Digitaldruck 438
Digitale Signatur 988
Digitalisierung 50
Digitalkamera 136
Auflösung 136
Dijkstra, Edsger W. 325, 594
DIMM-Module (RAM) 109
DIN 994
Papierformat 433
Schriftfamilien 426
dir, Windows-Befehl 222
Director 474
1-Bit-Bitmaps 485
3-D-Darsteller 493
3DPI-Xtra 493
Arbeitsoberfläche 476
Audio per Lingo steuern 510
Auwahl 487
Besetzung 477
Bitmap-Darsteller 483
Bitmaps importieren 492
Bühne 476
Darsteller 483
Darsteller importieren 491
Darsteller-Optionen 483
Drehbuch 478
Drehbuch-Elemente 479
Drehwerkzeug 487
Eigenschafteninspektor 481
Einfache Animation 479
Farbeffekte 482
Farbtiefe 484
Felddarsteller 490
Feldwerkzeug 488
Filmschleifen 494
Filmsteuerung 477
Formwerkzeuge 487
Frame-Markierungen 497
Frames 479
Geschichte 475
Import 491
Kanäle, Drehbuch 478
Keyframe 480
Lingo 494
Malfenster 483
Palettenkanal, Drehbuch 478
Projekt exportieren 511
Registrierpunkt 484
Rollover-Effekt 503
Skript-Darsteller 497
Skriptkanal, Drehbuch 479
Sound einsetzen 493
Sound importieren 493
Soundkanal, Drehbuch 479
Sprite-Eigenschaften 482
Sprites 479
Steuerpult 477
Tempokanal, Drehbuch 478
Textdarsteller 489
Textwerkzeug 487
Tweening-Optionen 481
Übergangskanal, Drehbuch 479
Vektordarsteller 490
Video per Lingo steuern 511
Werkzeugpalette 486
Disc-at-once 130, 994
Disjunktion, logische 57
Diskettenlaufwerk 128
diskrete Mengen 50
Diskussionsforum, PHP-Skript 851
display, CSS-Angabe 767
Division, Operator 235
DMA-Kanal 118, 994
DMA-Kanal, Direct Memory Access 118
DNA-Computer 42
DNS 994
BIND, Serversoftware 608
Master- und Slave-Nameserver 609
Subdomain 607
Top-Level-Domain 607
Zone 607
DNS (Domain Name System) 606
DNS-Server, Windows Server 2003 646
do/while()-Schleife 242
DocBook 686
Dock
Dock-Menü MacOS X Dock-Menü 210
MacOS X 208, 210
DOCTYPE, Verarbeitungsanweisung, HTML 721
DOCTYPE-Angabe
XML 696
Document Object Model → DOM
Document Type Definition → DTD
document.all, Internet Explorer-Eigenschaft 906
document.bgColor, JavaScript-Eigenschaft 868
document.forms, JavaScript 862
document.images, JavaScript 877
document.layers, alter Netscape-Layer-Zugriff 908
document.write(), JavaScript-Methode 857
Dokumentation 335
Administratorendokumentation 336
Anwenderdokumentation 336
Entwicklerdokumentation 335
Dokumentation, Software-Engineering, Javadoc 321
Dokumentstruktur
HTML 720
DOM 716, 994
Baumstruktur anzeigen, JavaScript-Anwendung 899
CSS-Formate ändern 902
Document-Objekt 716
Dokumenthierarchie ändern 904
DOMParser 716
für XML 716
getChildNodes(), Methode 717
in JavaScript 897
Kindknoten 717
Klassen importieren 716
Knoten-Eigenschaften 898
Knotentypen 717

Parser 716
Praktische Anwendung, JavaScript 902
Textknoten 898
Domain Name System 606
Domain Name System (DNS)
BIND-Nameserver 610
Round-Robin-Verfahren 614
Domain-Namen, Schema 607
DOMParser, Java-Klasse 716
Doppelte Anführungszeichen, Perl 271
DOS 150
DoS → Denial of Service
DOSKEY 221
Double Buffering, Animationstechnik 309
double, C-Datentyp 233
DOUBLE, SQL-Datentyp 375
DoubleWord (DWord) 994
DoubleWord, 32 Bit 70
Download, HTML-Hyperlinks 737
Drag and Drop 155
Drahtlose Netzwerke 564
Arten 565
Gründe für den Einsatz 564
Wireless LAN 565
Drahtlose Schnittstellen 122
Drain, Stromausgang des Transistors 74
DRAM 108
drawLine(), AWT-Methode 306
drawOval(), AWT-Methode 306
drawPolygon(), AWT-Methode 306
drawRect(), AWT-Methode 306
drawString, AWT-Methode 308
Dreamweaver 790
Bibliothek 799
Bilder einfügen 794
Code-Ansicht 791
CSS 801
DHTML erstellen 802
Dokument einrichten 790
Entwurfsansicht 791
Flash einbetten 796
Formate 791
Formulare 797
Frames erstellen 798
Layers erstellen 802
Layoutansicht 796
Listen 792
Navigationsleisten erstellen 795
Rollover 795
Tabellen 794
Textoptionen 791
Verhalten 802
Vorlagen 800
Website-Verwaltung 798
Zeitleiste 803
Drei Amigos 339
Drei-Wege-Handshake, TCP 603
DROP DATABASE, SQL-Befehl 374
DROP TABLE, SQL-Befehl 374
Druck
Ausschießen 439
Computer to Plate 439
Digital 438
Offset 438
Proof 439
Druckauflösung 392
Drucker 140
GDI-Drucker 141
im Netzwerk freigeben, Windows 645
Kugelkopfdrucker 140
Laserdrucker 141
LED-Drucker 141
Matrixdrucker 140
Nadeldrucker 140
Schriftarten 142
Thermosublimationsdrucker 141
Thermotransferdrucker 141
Tintenstrahldrucker 140
Treiber 141
Typenraddrucker 140
Druckraster 392
Rasterwinkel 392
Druckserver, Print-Server 555
DSL 573, 994
ADSL 573
Anschließen 574
Einrichten, Mac OS X 638
SDSL 573
über Fernsehkabel 574
über Sattelit 574
DSSS, WLAN-Technik 565
DTD 696, 994
Alternativen angeben 698
Attributdeklaration 702
Attributnotwendigkeit 704
Attributtypen 702
definieren 696
Elementdeklaration 698
Entites aus externen Dateien 705
Entities deklarieren 704
Externe Entity-Deklarationen 705
Häufigkeitsangabe 699
Klammern 698
DTP 421, 994
Ausschießen 439
Preflight 438
Seite einrichten 438
Texterfassung 438
DTP-Punkt, Schriftgröße 422
du, UNIX-Befehl 186
Dualsystem 49, 65
in Dezimal umrechnen 67
in Hexadezimal umrechnen 69
in Oktal umrechnen 68
Dualzahlen, mit Vorzeichen speichern 71
Duplex, Photoshop-Farbmodus 398
duplicateMovieClip(), ActionScript-Methode 967
Durchlicht-Scanner 135
Durchschuss, Typographie 424
DV-Camcorder 459

DVD 132
beschreibbare 133
Video auslesen 460
DVD+R 133
DVD+RW 133
DVD-R 133
DVD-RAM 133
DVD-ROM 132
Dateiformat 133
Geschwindigkeit 132
DVD-RW 133
DVI, Dateiformat 526
dvips, Dienstprogramm 526
DWord → DoubleWord, 32 Bit
Dynamic HTML 896
Dynamic RAM → DRAM

E

each(), PHP-Funktion 821
Ebenen
Flash 917, 930
Photoshop 407
spezielle, Photoshop 410
Ebenenmaske, Photoshop 409
echo(), PHP-Anweisung 817
Echte Obermenge 62
Echte Teilmenge 62
Eclipse 318
Installation 318
Java-Entwicklung mit 319
Perspektiven 319
Refactoring 322
Software-Engineering mit 318
Sprachpakete 318
ECMA 994
ECMAScript 855
EDO-RAM 109
EDV → Elektronische Datenverarbeitung
Effizienz der CPU 104
EIDE 120, 994
anschließen 120
im Vergleich zu SCSI 120
Ein-/Ausgabe 146
C 248
Dateien, Java 263
Dateien, Perl 274
Dialogverarbeitung 147
Fehler 255
Java 254
Lochkarten 146
Perl 266
Stapelverarbeitung 147
Terminal 147
Timesharing 147
Einfache Anführungszeichen, Perl 271
Einfügeabfrage 373
Eingabe
aus Dateien, C 249
Konsole, C 248
Perl 266
String, C 248
Zeichen, C 248
Eingabeaufforderung 995
UNIX 173
Windows 221
Eingabeaufforderung → Konsole
Eingabeeinheit 97
Eingabegeräte 123, 134
Digitalkamera 136
Maus 134
Scanner 135
Tastatur 134
Eingabesteuerung, durch das Betriebssystem 145
Eingabeumleitung
in UNIX-Shells 178
Windows 221
Eingabevervollständigung 176
Einsteckkarten 119
einbauen 119
Einwegverschlüsselung 988
Einzeltabellendatenbank 356
Definition 357
Filterung 358
Funktionen 358
Grenzen 358
Sortieren 358
Suchen 358
EJB → Enterprise Java Beans
El Torito (Booten von CD) 201
Elektrische Datenverarbeitung 25
Elektrische Geräte 35
Elektrizität 35
Elektromechanik 35
Elektronenröhre 36
Elektronik 35
Elektronische Datenverarbeitung 25
Elektronische Geräte 35
Element einer Menge 61
Eliza 84
else, C 240
elsif, Perl 273
Elternklasse 260
Emacs, Texteditor 195
Befehlseingabe 196
Befehlsschreibweise 196
Buffer 196
erweiterte Funktionen 197
Fenster wechseln 196
löschen 197
markieren 197
Modi 196
Navigation 197
Speichern 197
Suchfunktionen 197
Text ersetzen 197
E-Mail
Attachment 622
Entwicklung 540
Hoaxes 982
HTML-Hyperlinks auf 738
IMAP-Protokoll 624
Kettenmails 982
Multipart-Nachrichten 622
POP3-Protokoll 623
RFC-822-Nachricht 621
SMTP-Protokoll 619
Spam 980
Verschlüsselung 988
E-Mail-Protokolle 619
E-Mail-Server 555
Embedded Systems 99
Endlicher Automat 85
Endlosschleifen 283
Endung (Dateiname) 169, 180
Engelbart, Douglas 150
ENIAC, Röhrenrechner 36

Enterprise Java Beans 675, 995
Client-Interface 676
Deployment 678
Deployment-Deskriptor 679
Entity-Beans 676
Home-Interface 676
Implementierung 676
Message Driven Beans 676
Session-Beans 676
Vorteile 676
Enterprise-Anwendung 995
Enterprise-Anwendungen 674
Entity, Datenbank 357
Entity-Referenzen
HTML 723
XML 692, 704
Entwicklerdokumentation 335
Entwicklungsprozess, Software 336
Agile Entwicklungsprozesse 338
Extreme Programming 338
Unified Process 336
Entwicklungsumgebung, integrierte → IDE
Entwurf, Software-Engineering 333
Schnittstellen 334
Standalone-Systeme 333
Verteilte Systeme 333
Entwurfsmuster 345
Absicht 346
Alias 346
Beispiel Singleton 346
Bestandteile 345
Beteiligte 346
Codebeispiele 346
Einordnung 345
Einsatzbeispiele 346
Erzeugungsmuster 345
Implementierung 346
Konsequenzen 345, 346
Lösung 345
Motivation 346
Name 345
Problem 345
Querverweise 346
Singleton (als Beispiel) 346
Struktur 346
Strukturmuster 345
Verhaltensmuster 345
Verwendungszweck 346
Zusammenspiel 346
ENUM, SQL-Datentyp 376
EPOCH 156
EPROM 110
EPS, Dateiformat 526
Unterschiede zu PostScript 527
eq, Perl-Operator 272
equals(), Java-Methode 256
Ereignisanzeige, Windows 225
Ereignisbehandlung 303
Java, AWT 313
Ereignisprozeduren, Lingo 495
Ereignisse, AWT 304
Erweiterung (Dateiname) 169, 180
Erzeugungsmuster 345
Escape-Sequenz 995
Escape-Sequenzen 185
C 231
in RegExp 292
etc, UNIX-Verzeichnis 165
Ethernet 558, 995
10 Base 2-Standard 560
10 Base 5-Standard 561
10 Base T-Standard 562
100 Base T-Standard 562
1000 Base FL-Standard 563
1000 Base TX-Standard 563
Bridge 562
Crosslink-Kabel 562
Entwicklung 40
Hardware 560
Hardware-Adresse 558
Hub 562
Koaxialkabel 560
MAC-Adresse 558
Switch 562
Thicknet Coaxial 561
Thinnet Coaxial 560
Twisted-Pair-Kabel 561
Vorläufer ALOHANet 559
Ethernet-Kabeltypen 560
Euklidische Geometrie 80
EVA-Prinzip 95
Event Handler, Lingo 495
Event Handling 303
Event-Handler, ActionScript 954
Exabyte 70
Exceptions
auslösen 263
FileNotFoundException 264
IOException 264
Java 263
Exceptions, Java 255
IOException 255
exec(), C-Funktion 297
executeQuery(), JDBC-Methode 383
exit, UNIX-Befehl 177
Exklusiv-Oder 58
Expert-Zeichensatz 426
explode(), PHP-Funktion 821
Exploits 982
Explorer, Ordneransichten, Windows 220
Explorer, Windows-Dateimanager 219
Exponentialschreibweise 72
C 234
Exponentielle Komplexität 84
ext3, Linux-Dateisystem 186
extends, Java-Vererbung 260
extends, PHP-Schlüsselwort 829
Extensible Markup Language → XML
Extreme Programming 338
Eigenschaften 338
Programmieren in Paaren 338
Test-first-Verfahren 338, 351

F

Fachinformatiker 27
Anwendungsentwicklung 28
Projektarbeit 326
Prüfung 28
Systemintegration 28
Fahrrad, Flash-Projekt 939
Fakultät 281

Fallentscheidungen
Bedingung 239
C 239
durch logische Operatoren, Perl 272
in Shell-Skripten 191
switch/case 241
unless, Perl 273
Fallentscheidungs-Operator 238
Falsche Aussagen 54
false, Java 256
Farbaddition 50, 389, 995
Farbbalance, Photoshop 404
Farbe 388
CMYK 389
Color-Management 388
Farbaddition 389
Farbmodelle 389
Farbseparation 390
Farbsubtraktion 389
Farbtiefe 391
HSB-Modell 390
im Webdesign 773
Komplementärfarben 389
Körperfarben 389
Lab-Modell 390
Pigmentfarben 389
Primärfarben 389
Prozessfarben 390
Reflektion und Absorption 388
RGB 389
Sättigung 390
Vierfarbdruck 389
Volltonfarben 390
Farben
Ändern per JavaScript 868
in HTML 734
Farbkanal, Bild 50
Farbkanäle 995
Farbkanäle, Photoshop 401
Farbkorrektur 403
Farblaserdrucker 141
Farbmodelle 389
Farbmodus, Photoshop 398
Bitmap 398
CMYK 399
Duplex 398
Graustufen 398
Indizierte Farben 398
Lab 399
Mehrkanal 399
RGB 398
Farbpalette 398
Websichere Farben 773
Farbseparation 390, 995
Einstellungen, Photoshop 399
UCR 390
Farbsubtraktion 50, 389, 995
Farbtabelle 398
Farbtiefe 391, 995
der Grafikkarte 137
Director 484
Farbtiefe, Bild 50
Farbverlauf, Photoshop 406
Farbverläufe
Illustrator 416
Fast Ethernet 562
FAT 995
FAT 12, Dateisystem 215
FAT 16, Dateisystem 215
FAT 32, Dateisystem 216
FAT-Dateisysteme
FAT 12 215
FAT 16 215
FAT 32 216
fclose(), PHP-Funktion 836
FDDI 995
FDDI, Netzwerk 568
Fenster, programmieren 311
Fensterbedienung
MacOS X 208
Windows 218
feof(), PHP-Funktion 836
Festbreitenschrift 425
Festplatte, belegten Platz ermitteln (UNIX) 186
Festplatten 125
Anschlüsse 120
Cache 128
CHS-Adressierung 126
Defragmentierung 126
EIDE 120
formatieren 186
Geschwindigkeit 128
Konstante Winkelgeschwindigkeit 128
LBA-Adressierung 126
Mittlere Zugriffszeit 128
Partitionierung 126
RAID 127
SCSI 121
Festplattengrößen 70
Festwertspeicher → ROM
fg, UNIX-Befehl 175
fgets(), C-Funktion 249
fgets(), PHP-Funktion 836
FHSS, WLAN-Technik 565
FIFO 995
FIFO → First In, First Out
File Type ID, HFS 213
FileMaker, Datenbank 365
FileNotFoundException, Java 264
File-Server 554
fillOval(), AWT-Methode 306
fillRect(), AWT-Methode 306
Finder
MacOS X 208, 210
Ordneransichten, MacOS X 210
finger, UNIX-Dienstprogramm 172
Firewall 975
Bastion-Host 985
iptables 985
Firewalls 985
FireWire 122, 995
Fireworks 387, 776
Animierte GIFs 785
Arbeitsoberfläche 777
Bild austauschen, Verhalten 788
Bildauflösung 778
Bildgröße 778
Ebenen 777
GIF speichern 782
Image Maps 787
JPEG speichern 781
Navigationsleisten 789
PNG speichern 784
Popup-Menüs 789

Rollover-Verhalten 787
Segmente 786
Symbole 786
Verhalten 787
First In, First Out 285
Flachbettscanner 135
Flags, des virtuellen Prozessors 90
Flash 915
ActionScript 952
Aktionen-Palette 953
Animationstechniken 928
Ansicht verschieben 928
Arbeitsumgebung 916
Auswahlpfeil 918
Automatisches Ausrichten 918
Bandbreiten-Profiler 969
Begradigen 922
Bewegungs-Tweening 932
Bewegungs-Tweening am Pfad entlang 934
Bibliothek 938
Bitmaps importieren 944
Bitmaps vektorisieren 945
Bitmaps zerlegen 944
Dokumenteigenschaften 916
Drehen und Neigen 920
Ebenen 917, 930
Ebenen-Ansicht 931
Einzelbild-Animation 929
Ellipsenwerkzeug 921
Ereignis-Sound 943
Exportieren 946
Fahrrad, Animationsworkshop 939
Farbeffekte 942
Farbeffekte per ActionScript 966
Farbeimer 923
Farben 923
Farbtöne-Palette 924
Film abspielen 931
Film anhalten per ActionScript 953
Filme in HTML einbetten 948
Filme laden, ActionScript 956
Filme vorausladen 969
Form-Tweening 936
Formtweening, Formmarken 937
Formulare per ActionScript 964
Frames markieren 930
Frames verschieben 930
Freihandwerkzeug 922
Füllung sperren 923
Füllungen 923
Füllungstransformation 924
Glätten 922
Grafik, Symboltyp 938
Hand-Werkzeug 928
Hyperlinks per ActionScript 963
Importieren 942
in Dreamweaver einbetten 796
Instanzen 938
Instanzen erzeugen, JavaScript 967
Instanzen, Farbeffekte 942
Interaktivität 952
Keyframes 929
Lasso 918
Linienwerkzeug 921
Lupe 928
Magnet 918
Malmodus 922
Maskenebenen 935
Medienintegration 942
Movieclip, Symboltyp 938
Movieclip-Hierarchie 957
Movieclips einfärben, ActionScript 965
Movieclips steuern, ActionScript 957
Navigation, ActionScript 955
Objekte gruppieren 917
Pfade 919
Pinsel 922
Pipette 928
Preloader 969
Radiergummi 922
Rechteckwerkzeug 921
Schaltfläche, Symboltyp 938
Schaltflächen verwenden 954
Schaltflächen, Dauerfeuer 964
Schaltflächen-Aktionen 954
Skalieren 920
Sound importieren 943
Sound verwenden 943
Sound, Ereignis 943
Sound, Stream 943
Soundeigenschaften 943
Spielautomat, Beispielanwendung 959
Streaming-Funktion 969
Stream-Sound 943
SWF-Dateien exportieren 947
Symbole 937
Symbolinstanzen 938
Symboltypen 938
Szenen 955
Text zerlegen 936
Texteinstellungen 926
Textwerkzeug 925
Tintenfass 924
Transformations-Palette 920
Transformieren 919
Transformieren, Füllungen 924
Tweening, Bewegung 932
Tweening, Form 936
Vektorisieren von Bitmaps 945
Verbreitung 915
Veröffentlichen-Funktion 951
Verzerren 920
Video importieren 945
Werkzeugoptionen 918
Zeichenfunktionen 917
Zeichenstift 919
Zeichnen per ActionScript 968
Zeitleiste 929
Zeitleiste, Frames manipulieren 930
Zeitleistenoptionen 932
Zwiebelschaleneffekt 932
Flash-EPROM 110
Fließkomma-Literale 234

Fließkommazahlen 72
Exponentialschreibweise 72
Exponentialschreibweise, C 234
Flip-Flop (Schaltung) 78
float, C-Datentyp 233
FLOAT, SQL-Datentyp 375
Floating Point Numbers → Fließkommazahlen
FLOPS (CPU-Geschwindigkeit) 104
Flussdiagramm 81
Flusskontrolle 236
FM-Synthese (MIDI) 142
font-family, CSS-Angabe 766
font-size, CSS-Angabe 766
font-style, CSS-Angabe 766
font-weight, CSS-Angabe 766
fopen(), C-Funktion 249
fopen(), PHP-Funktion 835
for()-Schleife 243
for, Befehl, in Shell-Skripten 191
foreach, Perl-Schleife 269
fork(), C-Funktion 296
fork(), UNIX-Systemaufruf 158
Formale Logik 53
Formatieren, Datenträger 186
Fortran 45
Fortran, Programmiersprache 45
Foto-Multiplier (Trommelscanner) 136
FP-RAM 109
fprintf(), C-Funktion 249
fputs(), PHP-Funktion 838
Fraktur 428
Frame, AWT-Klasse 304, 311
Frames
in Dreamweaver 798
Schlüsselbilder 929
Frames, Datenpakete 559
Französische Renaissance-Antiqua 426
FreeBSD 148
FreeHand 387
Frege, Gottlob 53
Freie Software 170
Fremdschlüssel, RDBMS 359
Frequency Hopping 566
Frequenz, Audio 50
Front Side Bus 103
FSB → Front Side Bus
fscanf(), C-Funktion 249
fsck, UNIX-Befehl 186
FTP 617, 995
ASCII-/Binärmodus 619
Befehle 618
Clients 617
HTML-Hyperlinks auf 737
FULLTEXT, SQL-Schlüsselwort 377
function, JavaScript-Schlüsselwort 864
function, PHP-Schlüsselwort 823
Funk- Schnittstelle 123
Funktion 46, 995
Rückgabewert, C 231
Funktionale Programmiersprachen 48
Funktionen
Argumente 243
aufrufen 244
C 243
Call by Reference 245
Call by Value 245
iterative 282
Parameter 243
rekursive 281
Rückgabewert 243
SQL 379
Funktionsaufrufe, C 231
Fußgesteuerte Schleifen 242
fvwm2, Window-Manager 198

G

Gamma, Erich 318, 345
GAN 995
GAN, globales Netz 551
Ganze Zahlen 62
Gästebuch, PHP-Skript 842
Gate, Steuerungseingang des Transistors 74
gates (Gatter) → Logikschaltungen
Gates, Bill 39, 149
Gatter → Logikschaltungen
gcc, Befehl 229
UNIX-Besonderheiten 229
GCC, C-Compiler 228
starten 229
ge, Perl-Operator 272
Gebrochene Schriften 428
Gerätedateien 163, 164
Gerätetreiber 44, 145, 154, 636
Blockgeräte 154
für Drucker 141
installieren, Windows 223
Netzwerk, Windows 639
Zeichengeräte 154
Gestaltung
Anschnitt 436
Elemente anordnen 434
Goldener Schnitt 434
Grundregeln 433
Layout 435
Linien 436
Optische Mitte 435
Ornamente 437
Papierformat 433
Rahmen 436
Raster 437
Regeln im Webdesign 772
Schmuckelemente 437
Gestaltungsraster 437, 995
GET, HTTP-Befehl 627
zum HTML-Formularversand 750
getchar(), C-Funktion 248
getDate(), JavaScript-Methode 874
getDay(), JavaScript-Methode 875
getElementById(), JavaScript-Methode 897
getElementsByTagName(), JavaScript-Methode 897
getFullYear(), JavaScript-Methode 874

gethostbyname(), Systemaufruf 666
getHours(), JavaScript-Methode 875
getMinutes(), JavaScript-Methode 875
getprotobyname(), Systemaufruf 665
gets(), C-Funktion 231, 248
getSeconds(), JavaScript-Methode 875
getservbyname(), Systemaufruf 667
Getter-Methoden, Java 322
getURL(), ActionScript-Funktion 963
getYear(), JavaScript-Methode 874
Geviert, Typographie 424
GGT (Größter gemeinsamer Teiler) 277
GhostScript 526
GID (Group-ID) 172
von Prozessen 159
GIF, Bilddateiformat 531, 779
in Fireworks 782
GIFs, animierte, in Fireworks 785
Gigabit Ethernet 563
Gigabyte 70
GIMP, Bildbarbeitung 387
Glätten, Photoshop-Auswahl 400
Gleichheit 59
Gleichheit, Operator 236
Gleichung 54
Lösung 55
Gleichungen, lineare 55
Gleitkommazahlen
→ Fließkommazahlen
Global Area Network 551
global, PHP-Variablenmodifikation 824
Globale Variablen, C 234
GNOME, Desktop 157, 198, 200
Desk Guide 201
GNOME-Menü 201
Kontrollzentrum 201
Panel 201
Verknüpfungen erstellen 201
GNU Emacs, Texteditor 195
GNU General Public License 170
GNU Zip, Komprimierung 205
GNU-Projekt 170
go to, Lingo-Befehl 496
Goldener Schnitt 434
Gosling, James 251
GPL 170
Grafik 387
Grafikkarte 137
AGP 138
Auflösung 137
Farbtiefe 137
Geschwindigkeit 138
mit mehreren Monitoren 137
PCI 138
RAMDAC 138
Grafikprogramme 387
Grafische Benutzeroberfläche 146, 150, 157, 198, 996
Aqua, MacOS X 207
Drag and Drop 155
Ereignisse 303
Fensteranwendungen 311
GNOME 157, 198, 200
GtK+, Programmierumgebung 200
JFC 303
KDE 157, 198, 199
Luna (Windows XP) 217
Menüs programmieren 311
Nachrichten 303
Programmierung 303
Qt, Programmierumgebung 199
Quartz, Grafikbibliothek 207
Terminalfenster 173
UNIX 198
Widgets 303
Window-Manager 157
Windows 216
Windows 95 217
Windows 98 217
Windows XP 217
X Window 157, 198
Graphics, AWT-Klasse 304
Graphics2D, Java2D-Klasse 304
Graustufen, Photoshop-Farbmodus 398
Green Book (CDi) 130, 996
grep, UNIX-Befehl 184
Muster 184
GridLayout 312
Groß- und Kleinschreibung
HTML-Tags 722
igonorieren in RegExp 293
UNIX-Dateinamen 165
größer als, Operator 59, 236
größer oder gleich, Operator 237
Großrechner 38
Größter gemeinsamer Teiler 277
groupadd, UNIX-Befehl 188
Group-ID 172
von Prozessen 159
Grundlinie, Typographie 423
Grundlinienraster, InDesign 443
Grüner Balken (Unit-Test) 350
Gruppe
hinzufügen, UNIX 188
wechseln (Datei) 183
gt, Perl-Operator 272
GtK+, Grafikbibliothek 200
GUI → Grafische Benutzeroberfläche
Gültigkeitsbereich
Variablen, C 234
Variablen, Java 256
gunzip, UNIX-Befehl 205
gzip, Dateikomprimierung 205

H

Halbaddierer (Schaltung) 77
Halbleiter 37
Halteproblem (Berechenbarkeit) 81
Handschriftliche Antiqua 428
Handshake 996

Handshake, Modem-Kommunikation 571
Hard Links 166
Hardware
Ausgabegeräte 123, 136
BIOS 99
Bus 99, 115
Bus Mastering 118
Chipsatz 99
Digitalkamera 136
DMA-Kanal 118
Drucker 140
Eingabegeräte 123, 134
Grafikkarte 137
I/O-Basisadresse 118
IRQ 117
konfigurieren, Windows 223, 224
Maus 134
Monitor 138
Netzwerke 558
Onboard-Peripherie 99
Peripherie 123
Plug & Play 119
Prozessor 98
RAM 108
RAM-Speicher 98
Ressourcen 117
ROM-Speicher 98
Scanner 135
schematischer Aufbau 95
Schnittstellen 99
Soundkarte 142
Steuerung durch Betriebssystem 145
Tastatur 134
Zentraleinheit 98
Hardware Massenspeicher 123
Hardware-Interrupt 106
Harvard Mark I 36
Harvard Mark II 36
Hashes
PHP 820
Schlüssel 269
Hashes, Perl-Variablen 268
Hauptplatine → Mainboard
Hauptprogramm, main() als 47
Hauptspeicher 97
Hayes, Modem-Befehlssatz 571
Hayes-Befehlssatz 992
Header-Datei 248
Hello World 230
Helm, Richard 345
help, Windows-Befehl 157
Herunterfahren, Betriebssystem 190
Herunterfahren, Windows 219
Hewlett-Packard, HP UX, Betriebssystem 148
Hexadezimalsystem 66
in Dezimal umrechnen 68
in Dual umrechnen 69
Hexadezimalzahlen, C 234
Hex-Editoren 43, 527
HFS+, Dateisystem 212
HFS, Dateisystem 212
Data Fork 212
Resource Fork 213
HIER-Dokument, in UNIX-Shells 178
HIER-Dokumente
Perl 272
Variablensubstitution 272
Hintergrund (Prozesse) 175
Histogramm 404
History, UNIX-Shell 176
hitTest(), ActionScript-Methode 967
HKS, Sonderfarbskala 390
Hoaxes 982
Hollerith, Hermann 37
home, UNIX-Verzeichnis 165
Homecomputer 40, 149
Home-Verzeichnis 165, 168
Host-zu-Host-Transport, DDN-Modell-Schicht 546
Hot Spots 741
Hot-Plugging 122
HP UX, Betriebssystem 148
HPGL (Druckersprache) 141
hqx, Dateikomprimierung 213
HR/DSSS, WLAN-Technik 565
HSB-Farbmodell 390
HTML
Absatzausrichtung 725
Absätze 725
Absendeknopf 751
ActiveX-Objekte 754
Anker 738
applet-Tag 755
area-Tag 741
a-Tag 735
Attribute 720
Aufzählungen 730
Aufzählungszeichen wählen 731
AuswahlmenüHTML Formulare, Auswahlmenü 752
Beschreibung für Suchmaschinen 760
Bilder als Hyperlinks 740
Bilder einbetten 739
Blocksatz 726
Body 720
body-Tag 721
br /-Tag 722
b-Tag 727
Button 751
caption-Tag 743
Checkboxen 751
Client-Side-Image Maps 741
colgroup-Tag 747
Container-Tags 720
CSS 762
dd-Tag 733
Definitionsliste 733
dl-Tag 733
DOCTYPE-Angabe 721
Dokumentkopf 720
Dokumentkörper 720
Dokument-Standardfarben 734
Dokumentstruktur 720
Dokumenttitel 721
Dokumenttyp 721
Download-Hyperlinks 737
dt-Tag 733
Dummy-Frames 758
Einfach-Tags 720
E-Mail-Hyperlinks 738
embed-Tag 753
Entity-Referenzen 723

Farben 734
Farbnamen 734
fett 727
Flash-Filme einbetten 948
font-Tag 728
form-Tag 750
Formulardaten-Codierung 750
Formulare 749
Formulare, Absendeknopf 751
Formulare, Button 751
Formulare, Checkboxen 751
Formulare, Hidden-Felder 751
Formulare, Löschknopf 751
Formulare, Passwortfelder 751
Formulare, Radiobuttons 751
Formulare, Reset-Button 751
Formulare, Schaltfläche 751
Formulare, Submit-Button 751
Formulare, Textbereich 752
Formulare, Textfelder 751
Formulare, Versandmethode 750
Formularelemente 750
Formular-URL 750
Frames 755
Frames benennen 757
Frames verschachteln 756
Frames, Hyperlinks 756
Frames, Nachteile 758
Frames, randlose 757
frameset-Tag 755
frame-Tag 756
FTP-Hyperlinks 737
Geeignete Titel 721
Geschütztes Leerzeichen 725
GET, Formularversandmethode 750
Glossarliste 733
Groß- und Kleinschreibung 722
h1 bis h6, Tags 726
Head 720
head-Tag 720
Hidden-Formularfelder 751
Hintergrundbilder 742
html-Tag 720
Hyperlink, Zielangabe 757
Hyperlinks 735
Hyperlinks ins Web 736
Hyperlinks, in Framesets 756
Image Maps, client-seitige 741
Image-Maps 741
Image-Maps, serverseitige 741
img-Tag 739
input-Tag 751
i-Tag 727
Java-Applets einbetten 755
Kommentare 743
kursiv 727
Layer 769
Layout mit Tabellen 747
Layout-Tags 727
leere Tags 720
Links 735
Listen 730
li-Tag 730
Löschknopf 751
map-Tag 741
meta, Tag 759
meta-Tag 723
Meta-Tags 759
MIME-Types 753
Multimedia einbetten 753
nicht nummerierte Liste 730
nummerierte Liste 731
Nummerierungsart wählen 731
object-Tag 754
ol-Tag 731
Passwortfelder 751
Pfadangaben in Links 735
Pixel-GIF 748
Plug-in-Formate einbetten 753
POST, Formularversandmethode 750
pre-Tag 726
p-Tag 725
Radiobuttons 751
Randlose Frames 757
Refresh 759
Reset-Button 751
robots.txt-Datei 761
Schaltfläche 751
Schlüsselwörter für Suchmaschinen 760
Schriftart 729
Schriftfarbe 730
Schriftformatierung 728
Schriftgröße 728
script-Tag 856
Seite neu laden 759
Seiteninterne Links 738
select-Tag 752
Server-Side-Image Maps 741
Sonderzeichen 723
Strict, Dokumenttyp 721
Struktur-Tags 727
Style Sheets 762
style-Attribut 764
style-Tag 764
Submit-Button 751
Suchmaschinen-Informationen 760
Tabellen 742
Tabellen als Layoutmittel 747
Tabellen ausrichten 744
Tabellen, Gitternetzlinien steuern 746
Tabellen, Rahmenlinien steuern 746
Tabellen-Attribute 744
Tabellenbeschriftung 743
Tabellenbreite 744
Tabellenhöhe 744
Tabellenrahmen 744
Tabellen-Spaltenbreite 747
Tabellenzeilen 743
Tabellen-Zellabstand 744
Tabellenzellen 743
Tabellenzellen verbinden 745
Tabellenzellen-Attribute 745
Tabellenzellen-Ausrichtung 745
table-Tag 742
Tags 720
Tag-Verschachtelung 727

Target, Hyperlink 757
td-Tag 743
textarea-Tag 752
Textbereich 752
Textfelder 751
Textformatierung 722
Textmarken 738
th-Tag 743
title-Tag 721
Transitional, Dokumenttyp 721
tr-Tag 743
Überschriften 726
ul-Tag 730
Validatoren 721
Verschachtelte Listen 731
vorformatierter Text 726
Web-Palette 734
XHTML 719
XHTML-Besonderheiten 722
XHTML-Grundstruktur 721
Zeichenformatierung 727
Zeichensatz angeben 723
Zeilenumbruch 722
Zielangabe für Hyperlinks 757
HTML-Formulare
in Dreamweaver 797
per JavaScript modifizieren 862
Überprüfen per JavaScript 868
htpasswd, Apache-Dienstprogramm 659
HTTP 627, *996*
Header 628
Sitzung 627
Statuscodes 628
HTTP-Befehle 627
httpd.conf, Apache-Konfigurationsdatei 657
HTTP-Server 556
Hub 562
HUP, Signal 159
Hurenkind, Absatzfehler 433
Hybrid-CD 132, *996*
Hyperlinks, HTML 735
Hypertext 541, *996*

I
I/O → Ein-/Ausgabe
I/O-Basisadresse 118, *996*
I/O-Kanäle 177
IANA 579, *996*
IBM 40, 149
AIX, Betriebssystem 148
DB2 365
OS/2, Betriebssystem 150
IBM PC 40, 149
IBM WebSphere-Server 675
IC 36
ICMP *996*
ICMP-Protokoll 601
IDE 317
Borland 317
Eclipse 318
MS Visual Studio 317
UCSD-Pascal 317
IEEE *996*
IEEE 1394 995
IEEE 1394 → FireWire
IEEE 802.11 → Wireless LAN
IEEE, Netzzugangsverfahren 802.x 558
if(), C 239
if(), Fallentscheidung, nachgestellt, Perl 273
if, Befehl
in Shell-Skripten 191
if, Lingo 500
ifconfig, UNIX-Befehl 636
Illustrator 387
Angleichen, Werkzeug 419
Anordnung 419
Arbeitsoberfläche 412
Ausrichten 418
Auswahlwerkzeuge 413
Duplizieren 418
Effekte 417
Farben 415
Farbverläufe 416
Gruppieren 414
Konturen 415
Objekte anordnen 419
Objekte ausrichten 418
Objekte duplizieren 418
Pfade ändern 414
Pfade erstellen 412
Stapelreihenfolge 419
Stile 415
Stile speichern 418
Transformation 418
Transparenz 416
Zeichenstift 413
Zeichenwerkzeuge 415
Image Maps, in Fireworks 787
Image, AWT-Klasse 309
Bilddatei laden 309
Image, JavaScript-Klasse 878
Image-Maps 741
Imaginäre Zahlen 63
IMAP 624, *996*
Imperative Programmiersprachen 46
Implementierung, Software-Engineering 334
implode(), PHP-Funktion 821
import, Java-Anweisung 254
include(), PHP-Funktion 830
InDesign 437, 440
Absatz 443
Absatzformate 446
Arbeitsablauf 443
Auswahl 441
Blindtext generieren 438
Buntstift 442
Daten importieren 444
eingebettete Rahmen 445
Grundlinienraster 443
Konturenführung (Umfließen) 445
Paletten 443
Pfade 441
Preflight 447
Rahmen 442, 444
Rahmen verketten 445
Rahmen, eingebettete 445
Seitenlayout 443
Textwerkzeug 442
Transformation 442
Werkzeuge 441
Zeichen 443
Zeichenformate 446
Zeichenstift 441
Index, im RDBMS 359

INDEX, SQL-Schlüsselwort 377
indexOf(), Java-Methode 257
Indizierte Farben, Photoshop-Farbmodus 398
inet_aton, Netzwerkprogrammierung 666
Informatik 25
Angewandte 26
Bioinformatik 32
Medieninformatik 32
medizinische 32
Praktische 26
Studium 31
Technische 26
Theoretische 25
Wirtschaftsinformatik 32
Informatikkaufleute 28
Informationen
analoge 49
digitale 49
Infrarot-Anschluss 123
init-Prozess 158
Inner Join, SQL 379
durch WHERE ausdrücken 380
INNER JOIN, SQL-Klausel 380
InnoDB-Tabellen (MySQL) 364
inode 166
Input/Output → Ein-/Ausgabe
InputStreamReader, Java-Klasse 254
INSERT, SQL-Abfrage 380
Instanzen 252
erzeugen, Java 256
Instruction Table → Befehlstabelle
int
C-Datentyp 233
Funktionsdatentyp, C 230
INT, Signal 190
INT, SQL-Datentyp 375
Integer 996
Integer, Java-Klasse 279
Integer-Literale 234
Integrationsmanagement 328
Integrierter Schaltkreis 36
Intel 39, 100
Intel-Assembler 107
Interfaces
Java 261
Runnable, Java 262
Serializable, Java 262
Internet
Anwendungsprotokolle 615
Dateiübertragung 617
Denial of Service-Angriff 642
Dokumentation 540
Geschichte 539
Offizielle Einführung 541
RFC 540
Standards 540
Transportprotokolle 601
Vorläufer ARPANet 539
Zugang per Modem 570
Zugang über DSL 573
Zugang über ISDN 571
Internet Explorer 775
ActiveX 754
klassisches Objektmodell 906
Internet, DDN-Modell-Schicht 546
Internet-Schichtenmodell 545
Anwendungsschicht 547
Host-zu-Host-Transportschicht 546
Internetschicht 546
Netzzugangsschicht 546
Interpolation, Bildgröße 396
Interpreter 39, 45, 264
Inter-Prozess-Kommunikation 160
System V IPC 160
über Pipes 160
über Signale 160
Interrupt Request → IRQ
Interrupts, Hardware- 106
Intrusion Detection Systems 985
Network-IDS 985
Snort 985
Inverse Kinematik 996
Inverse Kinematik, 3-D-Animation 469
IO::Socket 668, 669, 670, 671
IOException, Java 255, 264
IP-Adresse
für Sockets 666
Subnet Mask 581
Teilnetzmaske 581
Zuweisen, Linux 637
Zuweisen, MacOS X 638
Zuweisen, Windows 639
IP-Adressen
Broadcast-Adresse 579
CIDR-Adressierung 581
CIDR-Berechnungen 583
IPv6 597
Klassen 578
Konzept 577
link local 580
Loopback 580
Netzwerkadresse 579
private 580
spezielle 580
Subnetting 582
Supernetting 582
VLSM 586
IPC → Inter-Prozess-Kommunikation
IP-Masquerading 596
IP-Protokoll
Adressverteilung 579
Datagramme 587
Default Gateway 589
Header 587
IPv6 597
Maximum Transmission Unit (MTU) 588
Multicasting 579
Network Address Translation (NAT) 596
Paketfragmentierung 588
Router 589
Routing 589
Routing-Protokolle 592
spezielle Adressen 580
TTL 592
IP-Routing
Beispiele 590
Routing-Tabellen 591
iptables 985
IPv4-Adressverteilung 579
IPv6 597

Adressaufbau 597
Datagramm-Header 599
Migration (Umstieg) 600
Motivation 597
Tunnelung 600
IPX/SPX, Netzwerkprotokoll 632
IrDA 123, 996
IRQ 117, 996
reserviert 118
Irreelle Zahlen 63
IS A-Beziehungen, OOP 342
is_float(), PHP-Funktion 826
is_int(), PHP-Funktion 826
is_null(), PHP-Funktion 826
is_string(), PHP-Funktion 826
ISA 119, 996
ISDN 571, 996
Anschließen 572
Kanalbündelung 572
Kanäle 572
ISO 996
ISO 9660 (CD-ROM-Format) 131
ISO-8859-Zeichensätze 520
Itanium, Prozessor 103
IT-Ausbildung 26
Fachinformatiker 27
Informatikkaufleute 28
IT-Systemelektroniker 28
IT-Systemkaufleute 28
Projektarbeit 326
Prüfung 28
Studienfächer 31
Iteration 282
ITS, Betriebssystem 147
IT-Sicherheit 971
IT-Systemelektroniker 28
IT-Systemkaufleute 28

J

J2EE 252, 675
Enterprise Java Beans 675
J2ME 252
J2SE 252
Jacobson, Ivar 339
Java 996
Installation 252
Virtuelle Maschine 251
Java Application-Server 675
Java Foundation Classes 303
Java Servlets 680
Beispiel 680
Java Virtual Machine (JVM) 251
java, Programm 254
Java, Programmiersprache 48, 251, 303
ActionListener 313
Animation, AWT 309
Applets 251, 304, 316
Ausnahmen 255, 263
Ausnahmen auslösen 263
AWT 304
Biddatei laden 309
Binäre Suche 285
boolean-Datentyp 256
BorderLayout 312
BufferedReader 254, 263
Button 311
byte-Datentyp 256
Callback-Methoden 302
Canvas 304
catch() 255
charAt()-Methode 257
class-Deklaration 254
CLASSPATH 252
Color 307
compareTo()-Methode 256
Connection, JDBC-Klasse 383
Dateien verarbeiten 263
Datenbankverbindung herstellen 383
Datentypen 255
drawLine()-Methode 306
drawOval()-Methode 306
drawPolygon()-Methode 306
drawRect()-Methode 306
drawString()-Methode 308
Eclipse-IDE 319
Eigenschaften 252
Ein- und Ausgabe 254
Einsatzgebiete 251
Enterprise Edition 675
equals()-Methode 256
Ereignisbehandlung 313
Exceptions 255, 263
executeQuery()-Methode 383
extends 260
false 256
fehlende Zeiger 257
FileNotFoundException 264
fillOval()-Methode 306
fillRect()-Methode 306
Frame 304, 311
Getter-/Setter-Methoden 322
Graphics2D, Klasse 304
Graphics-Klasse 304
GridLayout 312
Gültigkeitsbereich für Variablen 256
Image-Klasse 309
import-Anweisung 254
indexOf()-Methode 257
init(), Applets 316
InputStreamReader 254
Instanzen 252
Instanzen erzeugen 256
Integer-Klasse 279
Interfaces 261
IOException 255, 264
Java 280
*java.awt.** 304
*java.awt.event.** 304
*java.lang.** 257
*java.sql.** 382
*java.util.** 264
Java2D 304
Javadoc 321
*javax.swing.** 304
*javax.swing.event.** 304
*javax.swing.table.** 304
JDBC-Datenbankschnittstelle 382
JDBC-ODBC-Bridge 382
JDBC-Treiber laden 382
JFC 303
JFrame 304
Kapselung 253
Klassen 252
Kommentare 257
Kompilieren 253

Konstruktor 258
Label 311
lastIndexOf()-Methode 257
LayoutManager 312
length()-Methode 257
Lineare Suche 284
Listen 287
Listener 313
main()-Methode 254, 305
Math-Klasse 257
Menu 311
MenuBar 311
MenuItem 311
Methoden 252
MouseListener 313
MouseMotionListener 313
new 256
Object-Klasse 257
Objekte 252
Objektorientierung 252, 257
*org.xml.sax.** 713
*Package java.io.** 254
paint()-Methode 305
Panel 304, 311
parseInt()-Methode 279
print() 255
println() 255
private 258
Programme starten 254
protected 260
public 254, 258
readLine()-Methode 255
repaint()-Methode 309
ResultSet, JDBC-Klasse 384
RMI (Remote Method Invocation) 676
run()-Methode 300
Runnable-Interface 262, 300
SAX 713
SDK 251
SDK, Enterprise Edition (J2EE) 252
SDK, Micro Edition (J2ME) 252
SDK, Standard Edition (J2SE) 252
Serializable-Interface 262
Servlets 680
setColor()-Methode 307
setLayout() 312
show()-Methode 305
start()-Methode 300
Statement, JDBC-Klasse 383
static 254
String 255
String-Methoden 256
Strings vergleichen 256
String-Verkettung 255, 256
substring()-Methode 257
super 261
Swing 304
TextArea 311
TextField 311
this 258
Thread, Klasse 300
Threads 262, 300
throws-Klausel 263
true 256
try 255
try/catch-Blöcke 255
Überladen 259
Umwandlung von SQL-Datentypen 384
Unterschiede zu C 255
Unterstützung durch MacOS X 207
update()-Methode 309
Variablendeklaration 256
Vererbung 260
WindowListener 314
java.applet.*, Package 304
java.awt.*, Package 304
java.awt.event.*, Package 304
java.awt.Graphics, Klasse 304
java.io.*, Package 254
java.lang.*, Package 257
java.sql.*, Package 382
java.util.*, Package 264
Java2 Enterprise Edition 675
Java2 SDK 251
Enterprise Edition (J2EE) 252
Micro Edition (J2ME) 252
Standard Edition (J2SE) 252
Java2D-API 304
Java-Applets 304, 316
in HTML einbetten 755
init()-Methode 316
Javadoc 321
JavaScript 855, 996
+, Operator 859
Analoguhr, DHTML-Anwendung 909
angepasste Browserfenster öffnen 891
Ausgabe ins Dokument 857
außerhalb des Browsers 855
automatisierte Hyperlinks 890
Automatisierung in Dreamweaver 802
Bilder austauschen 876
Bilder vorausladen 878
Bildschirmgröße 893
Browserweiche 888
charAt(), String-Methode 869
CSS-Formate ändern, DOM 902
Date, Klasse 873
DHTML, browserübergreifende Lösungen 909
DHTML-Browserweiche 912
Diashow 883
document.all, Internet Explorer-Dokumentbaum 906
document.bgColor 868
document.forms, Formulare 862
document.images 877
document.layers, alte Netscape-Eigenschaft 908
document.write(), Methode 857
Dokumenthierarchie ändern, DOM 904
DOM, Objektmodell 897
DOM-Baumstruktur anzeigen 899
DOM-Knoteneigenschaften 898
DOM-Textknoten 898
ECMA-Standard 855
Event Handler 861

Externe Datei einbinden 857
Farben ändern 868
Fenstereigenschaften 891
Fenster-Methoden 895
Formular, Fokus setzen 870
Formulare überprüfen 868
Formularzugriff 862
Frames manipulieren 890
function, Schlüsselwort 864
Funktionen 864
Geschichte 855
getDate(), Methode 874
getDay(), Methode 875
getElementById(), DOM-Methode 897
getElementsByTagName(), DOM-Methode 897
getFullYear(), Methode 874
getHours(), Methode 875
getMinutes(), Methode 875
getSeconds(), Methode 875
getYear(), Methode 874
history-Objekt 889
Image, Klasse 878
in der Browser-History blättern 889
in HTML einbetten 856
indexOf(), String-Methode 869
Inhalte austauschen, Internet Explorer-Objektmodell 907
Internet Explorer-Objektmodell 906
klassisches Netscape-Objektmodell 907
lastIndexOf(), String-Methode 869
Layer-Inhalt austauschen, alle Browser 912
length, String-Eigenschaft 868
location-Objekt 890
Memory-Spiel 885
navigator-Objekt 888
open(), window-Methode 891
parseFloat(), Methode 860
parseInt(), Methode 859
prompt()-Methode 858
Reguläre Ausdrücke 870
Rollover, praxistaugliche Lösung 879
Rollover-Effekt 878
screen-Objekt 893
setTimeout(), Methode 875
Sicherheit 856
String, Klasse 868
String-Vergleiche 860
String-Verkettung 859
substring, String-Methode 869
Timeout 875
Vollbildfenster 894
Werbe-Popups verbannen 895
JavaServer Pages 681
javax.ejb.*, Java-Package 676
javax.swing.*, Package 304
javax.swing.event.*, Package 304
javax.swing.table.*, Package 304
Jaz-Laufwerk 128
JBoss, Java Application-Server 675
JDBC 997
JDBC, Java-Datenbankschnittstelle 382
JDBC-ODBC-Bridge 382
JFC 997
JFC (Java Foundation Classes) 303
JFrame, Swing-Klasse 304
Jobs, Steve 40, 149, 150
Johnson, Ralph 345
join, Perl-Anweisung 269
Join-Abhängigkeiten, RDBMS 364
Joins, SQL 379
Joliet 997
Joliet (CD-Format) 131
Joliet, Unterstützung auf dem Mac 132
Journaling-Dateisysteme 186
Joy, Bill 193, 251
JPEG, Bilddateiformat 532, 779
in Fireworks 781
JSP 681
JUnit 349
Beispiele 349
Grafische Oberfläche 350

K

Kabelanschluss als Internetzugang 574
Kanäle, Photoshop 401
Kapitälchen 425, 997
Kapselung 47, 253
Kartensteckplätze
AGP 119
ISA 119
PCI 119
PCMCIA 119
KDE 997
KDE, Desktop 157, 198, 199
Kontrollzentrum 200
Panel 200
Verknüpfungen erstellen 200
Kegelhöhe, Schriftgröße 422
Kernel 152
mach-Mikrokernel 152, 207
Mikrokernel 152
monolithischer 152
selbst kompilieren, Linux 205
Systemaufrufe 153, 155
Task Scheduler 153
Kernelmodus 153, 158
Kernighan, Brian 46, 148, 228
Kerning 997
Kerning, Typographie 424
Kettenmails 982
Keyframe 997
Keyframe, Director 480
Keyframes
3-D-Animation 470
Flash 929
keys, Perl 270
KI → Künstliche Intelligenz
Kildall, Gary 149
kill(), UNIX-Systemaufruf 159
KILL, Signal 159
kill, UNIX-Befehl 190

Kilobyte 70
Kindklasse 260
Klammern
in RegExp 293
Operatoren-Rangfolge verändern 238
Klassen 47, 252
abgeleitete 260
Elternklasse 260
Instanzen erzeugen, Java 256
Klassendiagramme (UML) 341
Klassentests → Unit-Tests
Klassizistische Antiqua 428
Kleinbildscanner 136
Kleincomputer 38
kleiner als, Operator 59, 236
kleiner oder gleich, Operator 237
Knuth, Donald E. 525
Koaxialkabel 997
KOffice, KDE-Office-Paket 199
Kommandozeile 997
Kommandozeile → Konsole
Kommandozeilenparameter, C 244
Kommentare
C 232
in HTML 743
Java 257
Perl 271
PHP 823
Kommunikation zwischen Prozessen 160
Kommunikationsmanagement 329
Kommunikationssteuerung, OSI-Schicht 544
Komplementärfarben 389
Komplexe Zahlen 63
Komplexität 997
exponentielle 84
logarithmische 84
O-Notation 83
polynomielle 84
quadratische 84
Komplexität von Algorithmen 82
Komplexitätsklasse 82
Komponentenmodelle 675
kompress, Typographie 424
Komprimierung 529
gzip 205
hqx 213
verlustbehaftete 529
verlustfreie 529
Konjunktion, logische 56
Konqueror, KDE-Browser 199
Konsole 146, 157
Ausgabe, C 248
Ausgabeumleitung 177
Befehle, Windows 222
Eingabe, C 248
Eingabeaufforderung 173
Eingabeaufforderung, Windows 221
Eingabeumleitung 178
Eingabevervollständigung 176
Pipes 178
Praktische Verwendung (UNIX) 171
Windows 221
Konsole, Starten, Windows 221
Konstante lineare Geschwindigkeit 130
Konstante Winkelgeschwindigkeit 128
Konstanten, symbolische 251
Konstruktor, Java 258
überladen 259
Kontextmenü, Mac OS X 211
Kontrollstrukturen 239
C 232
Fallentscheidung 239
foreach, Perl 269
Lingo 500
Schleifen 242
unless, Perl 273
Kooperatives Multitasking 154
Kopfgesteuerte Schleifen 242
Kopieren
Dateien, Mac OS X 211
Dateien, UNIX 180
Dateien, Windows 219
Dateien, Windows-Konsole 222
Koprozessor 100
Korn Shell (ksh) 174
Körperfarben 389
Kostenmanagement 329
Kritischer Pfad, Netzplan 330
Kryptoanalyse 987
Kryptografie 986
asymmetrische Verschlüsselung 987
Cäsar-Code 986
Digitale Signatur 988
Einwegverschlüsselung 988
Grundbegriffe 986
Message-Digest 988
ROT13 986
SSH 988
SSL/TLS 988
symmetrische Verschlüsselung 987
ksh (Korn Shell) 174
Kugelkopfdrucker 140
Künstliche Intelligenz 43
Kursiv, Schriftschnitt 425

L

Lab, Farbmodell 390
Lab, Photoshop-Farbmodus 399
Label, AWT-Klasse 311
Lamarr, Hedy 566
LAMP-System 810, 997
LAN 551, 997
Technische Lösungen 551
Wireless 564
Laserdrucker 141
Farbe 141
Lasso-Auswahl, Photoshop 400
Last In, First Out 285
Lastenheft 331
lastIndexOf(), Java-Methode 257
Lasttests 335
LaTeX 997
LaTeX, Satzsprache 525
Beispieldokument 525

Laufweite, Typographie 424
Laufwerke
Anschlüsse 120
magnetische 124
magneto-optische 124
optische 124
Laufwerksbuchstaben 167
Laufzeit 45
Laufzeitbibliothek, C 248
Lauschende Sockets 669
Layer-Animation, alle Browser 909
Layout
Anschnitt 436
Blindtext 437
Broschüren 435
Fachbücher 435
Kataloge 435
Linien 436
Prospekte 435
Rahmen 436
Romanseiten 435
Scribble 437
Seite einrichten 438
Tageszeitungen 435
Umfließen 436
Werbeanzeigen 436
Zeitschriften 435
LayoutManager
BorderLayout 312
GridLayout 312
LayoutManager, AWT 312
LBA (Festplattenadressierung) 126
LCD 997
LCD-Monitor 139
Funktionsprinzip 139
Nachteile 139
TFT 139
Vorteile 139
LDAP 997
le, Perl-Operator 272
Lead-In-Area (CD) 130
Lead-Out-Area (CD) 130
LED-Drucker 141, 997
Leere Menge 64
left, CSS-Angabe 769
Leibniz, Gottfried Wilhelm 34
Leichtgewichtige Entwicklungsprozesse 338
length(), Java-Methode 257
Lerdorf, Rasmus 810
Lernsoftware, CBT 452
less, UNIX-Befehl 183
letter-spacing, CSS-Angabe 766
Level-1-Cache 101
Level-2-Cache 101
LF, UNIX-Zeilenumbruch 516
Licht 388
Lichtfarben, RGB 50
Lichtwellen 388
Lichtwellenleiter 42
LIFO 997
LIFO → Last In, First Out
Ligaturen 425
LIKE, SQL-Klausel 378
Lilo, Linux-Bootmanager 202
Linear-Antiqua
serifenbetonte 428
serifenlose 428
Lineare Algebra 80
Lineare Geschwindigkeit, konstante 130
Lineare Gleichungen 55
Lineare Komplexität 83
Lineare Suche 83, 997
Java 284
line-height, CSS-Angabe 767
Lingo
Audio-Befehle 510
Cursor tauschen 504
Darsteller also Cursor 505
Darsteller-Skript 507
Eigenschaften 509
Eigenschaftslisten 510
Erstes Beispiel 494
Fallentscheidungen 500
Film-Skript 508
Frame-Skript 496, 507
globale Variablen 498
go to, Befehl 496
Gültigkeitsbereich von Variablen 498
if 500
Kontrollstrukturen 500
Listen 509
Navigationsbefehle 495
Positionen ermitteln 503
repeat-Schleife 500
Rollover-Effekt 503
Schleifen 500
Skript-Systematik 506
Sprite-Berührung testen 509
Sprite-Eigenschaften 503
Sprite-Skript 507
Strings verketten 502
Texte ändern 501
Variablen 498
Video-Befehle 511
Lingo, Director-Skriptsprache 494
Linien, Gestaltungselement 436
Linux 148, 149, 151, 169
$0, Systemvariable 174
/etc/passwd-Datei 172
/etc/shadow, Datei 173
als Server einrichten 647
Apache-Installation 655
Arbeitsverzeichnis anzeigen 181
auf NFS-Freigaben zugreifen 648
auf Windows-Server zugreifen 650
bash 174
Benutzerrechte 166
Bourne-Shell 174
cat-Befehl 183
cd-Befehl 181
chgrp-Befehl 183
Child-Prozess 159
chmod-Befehl 182
chown-Befehl 183
cp-Befehl 180
C-Shell 174
CUPS, Drucksystem 648
Daemon 189
date-Befehl 187
Dateibefehle 179
Dateibesitzer wechseln 183
Dateien kopieren 180
Dateien löschen 181

Dateien umbenennen 180
Dateien verschieben 180
Datei-Endungen 180
Dateigruppe wechseln 183
Dateinamen 165, 179
Dateinamen-Platzhalter 179
Dateisysteme 164
Datum und Uhrzeit ändern 187
Datum und Uhrzeit formatieren 187
Debian-Distribution 171
diff-Befehl 185
Distributionen 170
du-Befehl 186
Emacs, Texteditor 195
Escape-Sequenzen 185
exit-Befehl 177
fg-Befehl 175
finger, Dienstprogramm 172
fork(), Systemaufruf 158
fsck-Befehl 186
Gerätedateien 163, 164
GNOME 157, 198, 200
GNU-Projekt 170
Grafische Benutzeroberfläche 198
grep-Befehl 184
groupadd-Befehl 188
Group-ID 159, 172
gunzip-Befehl 205
gzip-Komprimierung 205
Hard Links 166
Hardware-Plattformen 169
Hardware-Unterstützung 202
HIER-Dokument 178
Home-Verzeichnis 165
ifconfig-Befehl 636
init-Prozess 158
inodes 166
Installation 201
Installation von Software 204
IP-Adresse zuweisen 637
KDE 157, 198, 199
Kernel kompilieren 205
kill(), Systemaufruf 159
kill-Befehl 190
Konfiguration 201
Korn Shell 174
less-Befehl 183
Lilo, Bootmanager 202
Links (Dateisystem) 166
Login 171
ls-Befehl 181
Macintosh im Netz bedienen 651
make-Befehl 205
man-Befehl 157
Mandrake-Distribution 171
mkdir-Befehl 182
mkfs-Befehl 186
more-Befehl 183
mount-Befehl 166, 185
mv-Befehl 180
MySQL-Installation 369
netatalk-Server 651
Netzwerkkonfiguration 635
Neustart 190
NFS 647
Pager 183
Paketmanager 204
Parent-Prozess 159
passwd-Befehl 188
Passwort ändern 188
patch-Befehl 185
path, Umgebungsvariable 175
pause(), Systemaufruf 159
Pfadangaben 165
PHP-Installation 811
Pipes 178
Prozessmodell 158
Prozessverwaltung 189
ps-Befehl 159, 189
pstree-Befehl 189
pwd-Befehl 181
RedHat-Distribution 171
Reguläre Ausdrücke suchen 184
rm-Befehl 181
rmdir-Befehl 182
root, Benutzer 159, 172
route-Befehl 637
rpm, Paketmanager 204
Runlevel 203
Samba-Server 649
SaX, X-Konfigurationsprogramm 199
Shell 157, 171
Shell-Ausgabeumleitung 177
Shell-Eingabeumleitung 178
Shell-Eingabevervollständigung 176
Shell-History 176
Shell-Skripte 190
shutdown-Befehl 190
Software installieren 204
Standardrouter einrichten 637
startx-Befehl 199
su-Befehl 177
SUSE-Distribution 171
Swap-Partition 162
Symbolic Links 166
System herunterfahren 190
System V Init 202
Systemprogramme 169, 179
tar-Dateien 205
Textbefehle 183
Textdatei anzeigen 183
Textdateien vergleichen 185
Texteditoren 192
Treiber im Kernel installieren 636
Umgebungsvariablen setzen 176
umount-Befehl 186
useradd-Befehl 188
User-ID 159, 172
Verwaltungsbefehle 185
Verzeichnis anlegen 182
Verzeichnis löschen 182
Verzeichnis wechseln 181
Verzeichnisbaum 164
Verzeichnisbefehle 179
Verzeichnisinhalt anzeigen 181
vi, Editor 193
virtuelles Terminal 173
wc-Befehl 185
Window-Manager 157
Wörter zählen 185
X Window 157, 198

XConfugurator, X-Konfigurationsprogramm 199
yast, SUSE-Konfigurationstool 204
Zugriffsrechte 166
Lisp, Programmiersprache 48, 198
Listen
C 286
Java 287
Lingo 509
Perl 268
qw//, Perl 271
listen(), Perl-Funktion 669
Listener, AWT-Ereignisbehandlung 313
Literal 997
Literale 234
Fließkommazahlen 234
Integer 234
Strings 235
Zeichen 235
Literale, Perl 270
Little-Endian-Architektur 992, 997
Little-Endian-Plattformen 527
loadMovie(), ActionScript-Funktion 956
Local Area Network 551
LocalTalk 630
localtime(), C-Funktion 250
Lochkarte 37
Lochkarten 146
Logarithmische Komplexität 84
Logical Link Control (LLC) 544
Logik
AND 56
Aussageformen 54
Aussagelogik 53
Aussagen 53
Definition 53
Disjunktion 57
Formale Logik 53
Gleichung 54
Konjunktion 56
Lügende Kreter 82
Mathematische Aussagen 54
Operatoren, C 235
OR 56
Prädikatenlogik 48, 53
Schlussfolgerung 55
Term 54
Umkehrschluss 56
Ungleichung 54
Verknüpfungen 55
Wahre und falsche Aussagen 54
Widerspruch 82
XOR 58
Logikschaltkreis 73
Logikschaltungen
Addierwerk 77
AND/OR-Aufbau durch Transistoren 75
Flip-Flop 78
Gattersymbole 75
Halbaddierer 77
mit einfachen Mitteln nachbauen 73
Multiplexer 76
NAND-Schaltung 75
NOR-Schaltung 75
NOT-Schaltung 74
RS-Flip-Flop 78
Speicherzelle 78
Volladdierer 77
XOR-Schaltung 76
Login 997
Login (UNIX-Anmeldung) 171
Logische Operatoren
C 235
Short-Circuit-Logik 273
Logische Programmiersprachen 48
Logische Schlussfolgerung 55
Logische Verknüpfungen 55
in Programmiersprachen 60
Logisches Nicht, Operator 235
Logisches Oder, Operator 235
Logisches Und, Operator 235
Logo, Programmiersprache 48
Logos 53
Lokale Variablen, C 234
Lokales Netz, Entwicklung 40
Lokalisierung
Mac OS X 212
long, C-Datentyp 233
LONGBLOB, SQL-Datentyp 376
LONGTEXT, SQL-Datentyp 376
Loopback, IP-Protokoll 580
Lorem Ipsum, Blindtext 438
Löschabfrage 373
Löschen
Dateien, UNIX 181
Dateien, Windows-Konsole 222
Verzeichnis, Windows 222
Lösung, einer Gleichung oder Ungleichung 55
Lösungsmenge 55
Lovelace, Ada 35
ls, UNIX-Befehl 181
LS-120-Laufwerk 128
lt, Perl-Operator 272
Luna, WinXP-Benutzeroberfläche 217
Lupe, Photoshop 397
LVALUE 237
Liste als 268

M

m, n-Relation, RDBMS 359
m//, Perl-Operator 291, 294
Mac 41
Mac OS 9, MacPerl 265
Mac OS X 148, 206
Alias 211
als Server einrichten 644
Anwendungsmenü 209
APIs 207
Apple-MenüApple-Menü Mac OS X 208, 209
Aqua 148, 207
Aqua, praktische Anwendung 208
Aqua-Fenster 208
Architektur 206
Benutzerverwaltung 212
Carbon 207
Classic-Umgebung 207

Cocoa 207
Darwin 207
Data Fork 212
Dateien kopieren 211
Dateien umbenennen 211
Dateien verschieben 211
Dateisysteme 212
Desktop 208
Dock 208, 210
DSL einrichten 638
File Sharing 644
Finder 208, 210
Geschichte 206
HFS+-Dateisystem 212
HFS-Dateisystem 212
hqx-Format 213
IP-Adresse zuweisen 638
Java 207
Konfiguration 211
Kontextmenü 211
Landeseinstellungen 212
Menüleiste 208
Monitore, Systemeinstellungen 212
Netzwerkkonfiguration 637
OpenGL, 3D-Grafikbibliothek 207
Ordneransichten 210
Papierkorb 210
Quartz, Grafikbibliothek 207
QuickTime 207
Resource Fork 213
Server nutzen 644
Startvolume einstellen 211
Systemeinstellungen 211
Terminal starten 208
UNIX-Komponenten 208
Windows File Sharing 644
Mac9, Perl-Unterstützung *265*
MAC-Adresse 558, *997*
mach-Mikrokernel 152
Macintosh 41, 150
BIOS 114
Firmware-Update 115
Leonardo, ISDN-Leonardo, Mac-ISDN-DFÜ 573
Mac OS X 206
Parameter-RAM 115
Serversysteme 643
MacPerl 265
Macromedia
Director 474
Dreamweaver 790
Fireworks 387, 776
Flash 915
FreeHand 387
Magnetbänder 125
Magnetbandspeicher 38
Magnetische Datenträger 124
Magnetbänder 125
Magnetscheiben 125
Magneto-optische Datenträger 124
Mail-Server 555
mailto, HTML-Hyperlink 738
main(), AWT-Methode 305
main(), C-Funktion 47, 230, 243
Kommandozeilenparameter 244
main(), Java-Methode 254
Mainboard 98
Chipsatz 99
Kartensteckplätze 119
Onboard-Peripherie 99
Mainframe 38
Majuskelziffern 426
make, UNIX-Befehl 205
Makefile 205
Makroviren 973
malloc(), C-Funktion 287
Malmodus, Photoshop 405
Malwerkzeuge, Photoshop 404
MAN 997
MAN, Stadtnetz 551
man, UNIX-Befehl 157
Mandrake, Linux-Distribution 171
Manetscheiben 125
Man-in-the-Middle-Angriff 983
Manuelle Datenverarbeitung 25
Marconi, Guglielmo 564
margin, CSS-Angabe 767
Mark I, Röhrenrechner 36
Mark II, Röhrenrechner 36
Maschinenbefehle 107
Maschinensprache 43
Massenspeicher 123
CD-ROM 129
Diskettenlaufwerk 128
DVD 132
Festplatten 125
Jaz-Laufwerk 128
LS-120 128
magnetische 124
magneto-optische 124
optische 124
Übersicht 124
Wechseldatenträger 128
Zip-Laufwerk 128
Master-Nameserver 609
Matching, Reguläre Ausdrücke 294
Math, Java-Klasse 257
Mathematische Aussagen 54
Mathematische Funktion, Undefiniertheitsstelle 82
Mathematische Variablen 54
Mathematischer Term 54
Matrixdrucker 140
Maus 134
MAX, SQL-Funktion 379
Maximum Transmission Unit (MTU) 588
md → mkdir, Windows-Befehl
MDI, Windows-Anwendungen 218
Mechanische Datenverarbeitung 25
Media Access Control (MAC) 544
Median, QuickSort 282
Mediävalziffern 426
MediaWiki, Wikipedia-Software 810
Mediengestalter 29
Bild/Ton 29
Medienberatung 29
Mediendesign 29
Medienoperating 29

Medientechnik 29
Prüfung 30
Medieninformatik 32
MEDIUMBLOB, SQL-Datentyp 376
MEDIUMINT, SQL-Datentyp 375
MEDIUMTEXT, SQL-Datentyp 376
Medizinische Informatik 32
Megabyte 70
Megapixel (Digitalkamera) 136
Mehrfachvererbung 261
Mehrkanal, Photoshop-Farbmodus 399
mem.h, C-Bibliothek 287
Memory Management Unit → MMU
Memory-Spiel, JavaScript-Anwendung 885
Mengen, diskrete 50
Mengenoperationen 60
Echte Teilmenge 62
Element 61
Leere Menge 64
Obermenge 62
Restmenge 64
Schnittmenge 63
Teilmenge 61
Vereinigungsmenge 64
Verknüpfungen 63
Menu, AWT-Klasse 311
MenuBar, AWT-Klasse 311
MenuItem, AWT-Klasse 311
Message-Digest 988
Message-Passing 303
METAFONT, TeX-Zeichensätze 525
Methoden 252
Callbacks 302
überladen 259
Metropolitan Area Network 551
Microsoft 39, 149
ASP.NET 808
MS-DOS 150
SQL-Server 365
Visual Studio 317
Windows 150, 213
Windows 2000 151
Windows 95 151
Windows 98 151
Windows Me 151
Windows NT 150
Windows XP 151
Windows, Versionsübersicht 214
Microsoft BizTalk Server 647
Microsoft Exchange Server 647
Microsoft Management Console (MMC) 225
Microsoft SQL Server 647
Microsoft Systems Management Server 647
MIDI 142, 997
FM-Synthese 142
in Wave-Editoren 455
Wavetable-Synthese 143
MIDI-Sequencer 454
Mikrokernel 152
mach 152, 207
Mikroprozessor 39, 98
Mikroprozessor → Prozessor
Mikrotypographie 429
Satzzeichen 429
Zahlen 430
MilNet 541
MIME 997
E-Mail-Nachricht 621
MIME-Multipart-Nachrichten 622
MIME-Nachrichtenheader 621
MIME-Type
HTML-Plug-in-Formate 753
XML-Dokumente 686
MIME-Types 621
im HTTP-Header 628
MIN, SQL-Funktion 379
Minicomputer 38
Minix, Lehrbetriebssystem 151
Minuskelziffern 426
MIPS (CPU-Geschwindigkeit) 104
MIPS, Prozessor 105
Mirroring (RAID) 127
MITS 39
Mittellänge, Schriftgröße 423
Mixed-Mode-CD 130
mkdir, UNIX-Befehl 182
mkdir, Windows-Befehl 222
mkfs, UNIX-Befehl 186
MMC (Microsoft Management Console) 225
MMU 102, 162, 997
Seitentabelle 162
MMX (CPU-Befehlserweiterung) 105
Mnemonics 44, 992
Modem 570
Anschließen 571
AT-Befehlssatz 571
Handshake 571
Hayes-Befehlssatz 571
Pulswahl 571
Tonwahl 571
Modularisierung von Programmen 46
Modularisierung, Programme 46, 243
Modularität, von UNIX 148
Module, Apache 653
Modulo, Operator 235
Modus, Photoshop 398
Moirée-Effekt, verhindern 392
Monitor 138
Bildwiederholrate 138
Konfiguration, Mac OS X 212
LCD 139
mehrere verwenden 137
Röhrenmonitor 138
Zeilenfrequenz 138
Monlithischer Kernel 152
more, UNIX-Befehl 183
MosTek 41
MosTek 6502, Prozessor 103
Motherboard → Mainboard
Motorola 68000, Prozessorfamilie 103
mount, UNIX-Befehl 166, 185
Mounten 998
Dateisysteme 166, 185
NFS-Freigaben 648

Mouse, ActionScript-Objekt 965
MouseListener 313
MouseMotionListener 313
MovableType, Blog-Tool 809
MOV-Befehl
beim virtuellen Prozessor 89
x86-Assembler 107
move, Windows-Befehl 222
Mozilla 775
MP3, Audio-Dateiformat 534, 998
MP4, Audio-Dateiformat 534
MPEG 998
MPEG, Video-Dateiformat 535
MS-DOS 150
Kommandozeile 221
MS-DOS-Anwendungen, unter Win32 215
MTU 588, 998
Multi Document Interface → MDI, Windows-Anwendungen
Multicasting, IP-Protokoll 579
MULTICS 147
Multimedia 451
Autorensysteme 474
CBT 452
Einsatzgebiete 452
Medientypen 451
Vebreitungskanäle 451
Multimedia-Datenbank 356
Multimedia-Produktion 451
Multipart-E-Mails 622
Multiplexer (Schaltung) 76
Multiplikation, Operator 235
Multiplikator
bei Rambus-RAM 109
der Taktfrequenz 103
Multi-Processing-Module (MPM), Apache 653
Multisession-CD 130
Multitasking 154, 158
kooperatives 154
präemptives 154
Unterstützung durch CPU 101
mv, UNIX-Befehl 180
MX-Record, BIND-Nameserver 615
my, Perl-Variablendeklaration 267
MyISAM-Tabellen (MySQL) 364
MySQL 368
Datentypen 375
InnoDB-Tabellen 364
Installation (UNIX) 369
Installation (Windows) 370
JDBC-Anbindung 382
Konfiguration (Windows) 370
MyISAM-Tabellen 364
MySQL Administrator 370
MySQL Query Browser 370
mysqli, PHP-Schnittstelle 849
mysql-Kommandozeilen-Client 372
phpMyAdmin 368
Sicherheitshinweise (UNIX) 369
Sicherheitshinweise (Windows) 371
Testdatenbank 845
Transaktionen 364
MySQL Connector/J, JDBC-Schnittstelle 382
MySQL Grant Tables 369
mysqli-Schnittstelle 849

N

Nachrichtenübermittlung 303
Nadeldrucker 140
Namensräume, XML 705
Nameserver
BIND 610
Master 609
Slave 609
NAND-Schaltung 75
Nassi/Shneiderman-Diagramm 81
NAT 998
NAT, IP-Protokoll 596
IP-Masquerading 596
Natürliche Zahlen 62
navigator, JavaScript-Objekt 888
ne, Perl-Operator 272
Nebenläufigkeit 295
Nessus 986
netatalk
afpd 651
atalkd 651
Konfiguration 651
papd 651
Verzeichnisse freigeben 651
netatalk, Serverdienst 651
NetBEUI 998
NetBEUI, Netzwerkprotokoll 633
NetBIOS, Windows-Netzwerk 633
Netscape 775
klassisches Objektmodell 907
Netscape-Palette 734
netstat, TCP/IP-Dienstprogramm 642
Routing-Tabellen anzeigen 591
Network Address Translation (NAT) 596
Network File System → NFS
Netzplan 329
kritischer Pfad 330
Netzwerk 537
Anwendungsgebiete 537
Client-Server 553
Denial of Service-Angriff 642
Drucker freigeben, Windows 645
Geschichte 539
Hardware 537
Konfiguration 635
Konfiguration, Linux 635
Konfiguration, MacOS X 637
Konfiguration, Windows 639
lokales, Entwicklung 40
Netzwerkprogrammierung 664
Peer-to-Peer 553
Protokolle 537
Server konfigurieren 643
Topologie 552

Verkabelung 537
Zentralisierungsgrad 553
Netzwerk, Linux
→ Gerätetreiber
Netzwerk, OSI-Schicht 544
Netzwerkclient, Definition 554
Netzwerke
ARCNet 569
ATM 568
drahtlose 564
Ethernet 558
FDDI 568
Funktionsebenen 543
GAN (Global Area Network) 551
Hardware 558
IEEE 802-Standard 558
Klassifizierung 550
LAN (Local Area Network) 551
Logical Link Control 544
MAN (Metropolitan Area Network) 551
Media Access Control 544
OSI-Referenzmodell 543
Protokolle 576
Reichweite 551
Schichtenmodelle 543
TCP/IP-Protokolle 576
Token Ring 563
WAN (Wide Area Network) 551
Zugang per Modem 570
Zugang über DSL 573
Zugang über ISDN 571
Netzwerkhardware 558
Ethernet 560
ISDN-Adapter 572
Modem 570
Netzwerkprogrammierung 664
Berkeley Socket API 664
IO::Socket 669
sockaddr_in 666
Sockets 664
Netzwerkprotokolle 537
AppleTalk 629
IPX/SPX 632
NetBEUI 633
Netzwerkserver, Definition 553
Netzwerktopologie 552
Baum 552
Bus 552
logische 552
physikalische 552
Ring 552
Stern 552
Netzzugang, DDN-Modell-Schicht 546
Netzzugangsverfahren
CSMA/CA 566
CSMA/CD 559
IEEE 802-Standards 558
Token-Passing 563
Neumann, John von 36, 96
Neuronale Netze 43
Neustart, Betriebssystem 190
Neustart, Windows 219
new, Java-Operator 256
Newsgroups 625
Hierarchie 626
NNTP-Protokoll 625
NeXT 206
NFS 647, 998
/etc/exports, Konfigurationsdatei 648
auf andere Server zugreifen 648
Starten 648
Verzeichnisse freigeben 648
NNTP 625, 998
Header 625
Normalformen, RDBMS 362
Normalisierung 998
Normalisierung, RDBMS 362
NOR-Schaltung 75
NOT-Schaltung 74
Novell NetWare 543
NSFNet 541
nslookup, TCP/IP-Dienstprogramm 642
NS-Record (DNS) 614
NTFS, Dateisystem 216
Nucleus, Blog-Tool und CMS 809
Null, im Stellenwertsystem 34
NULL, leerer C-Zeiger 250
NULL, SQL-Feldoption 377
Null-terminierte Strings 246
NURBS 998
NURBS, 3-D-Animation 470
Nyquist-Theorem 453, 998

O

Oberlänge, Typographie 423
Obermenge 62
Object Database Management Group → ODMG
Object Definition Language 366
Object Management Group 339
Object Management Group (OMG) 339
Object Query Language
→ OQL
Object, Java-Klasse 257
Objective-C, Programmiersprache 207
Objekte 47, 252
Objektorientierte Analyse 331
Objektorientierte Datenbank 356, 365
Abfragen 367
ODL 366
OQL 367
Objektorientierung 47, 252, 998
Eigenschaften 252
Elternklasse 260
Ereignisse 303
im Software-Engineering 325
Instanzen 252
Instanzen erzeugen, Java 256
Interfaces, Java 261
IS A-Beziehungen 342
Java 257
Kapselung 47, 253
Kindklasse 260
Klassen 47, 252
Konstruktor 258

Methoden 252
Nachrichten 303
Objekte 252
Perl 670
PHP 826
Überladen 259
Vererbung 47, 260
ODBC 998
ODBC, Microsoft-Datenbankschnittstelle 382
Oder-Schaltung → OR-Schaltung
Oder-Verknüpfung → OR-Verknüpfung
ODL 366
ODMG 366
OFDM, WLAN-Technik 565
Offsetdruck 438
OGG Vorbis, Audio-Dateiformat 534
Oktalsystem 66
in Dual umrechnen 68
Oktalzahlen, C 234
OMG → Object Management Group
Onboard-Hardware 99
onClipEvent(), ActionScript-Handler 965
Online-Dienste 541
O-Notation 998
O-Notation (Komplexität) 83
OODBMS 366
OOP → Objektorientierung
Open Database Connectivity → ODBC, Microsoft-Datenbank
open, Perl-Anweisung 275
opendir, Perl-Anweisung 275
OpenGL, 3D-Grafikbibliothek 207
OpenType-Schrift 440
Opera, Browser 775
Operatoren 236
- 235
! 235
!= 236
!~ 294
$, RegEx 293
% 235
&&, C 235
&, C 236
* 235
**, RegExp* 291
+ 235
+, Java-String-Verkettung 255, 256
+, RegExp 291
-, C 247
., Perl 272
/ 235
= 237
== 236
=~ 294
=~, Perl 276
? 238
?, RegExp 291
^, *C* 236
^, *RegExp* 291, 293
`` 297
| 236
|, RegExp 293
|| 235
~ 236
Addition 235
arithmetische 235
Backticks 297
Bit-Komplement 236
Bit-Verschiebung, links 236
Bit-Verschiebung, rechts 236
bitweise 236
Bitweise Exklusiv Oder 236
Bitweise Oder 236
Bitweise Und 236
C 235
Division 235
eq, Perl 272
Fallentscheidung 238
ge, Perl 272
Gleichheit 236
größer als 236
größer oder gleich 237
gt, Perl 272
kleiner als 236
kleiner oder gleich 237
le, Perl 272
logische 235
Logisches Nicht 235
Logisches Oder 235
Logisches Und 235
lt, Perl 272
m// 294
Mengen 60
Modulo 235
Multiplikation 235
ne, Perl 272
Perl 270
PHP 819
Post-Dekrement 238
Post-Inkrement 238
Prä-Dekrement 237
Prä-Inkrement 237
Rangfolge 238
Rangfolge durch Klammern ändern 238
s/// 294
Stringvergleiche, Perl 271
String-Verkettung, Java 255, 256
Subtraktion 235
tr/// 295
Ungleichheit 236
Vergleiche 59, 236
Wertzuweisung 237
opt, UNIX-Verzeichnis 164
Optionen, bei Systemprogrammen 175
Optische Datenträger 124
Optische Mitte 435
OQL 367
Oracle, Datenbank 365
Orange Book 998
Orange Book (CD-R, CD-RW) 130
ORDER BY, SQL-Klausel 379
Ordnungsdaten 355
org.w3c.dom.*, Java-Package 716
org.xml.sax.*, Java-Package 713
Ornamente, Gestaltungselement 437
OR-Schaltung 75
Aufbau mit Transistoren 75

mit einfachen Mitteln nachbauen 74
OR-Verknüpfung 56
OR-Vernüpfung, Wahrheitstabelle 58
OS/2, Betriebssystem 150
OSI-Referenzmodell 543, 998
Anwendungsschicht 545
Bitübertragungsschicht 544
Darstellungsschicht 545
Kommunikationssteuerungsschicht 544
Netzwerkschicht 544
Präsentationsschicht 545
Sicherungsschicht 544
Sitzungsschicht 544
Transportschicht 544
Vergleich mit der Praxis 545
Vergleich zum Internet-Schichtenmodell 545
OSPF 998
OSPF, Routing-Protokoll 594
Outsourcing 542
Overclocking → Übertakten

P

Paare, Extreme Programming 338
packet switching 538
padding, CSS-Angabe 767
Page Fault (Speicher) 162
Page File (Auslagerungsdatei) 162
Pager (seitenweise anzeigen) 183
Paging (Speicher) 102, 162
paint(), AWT-Methode 305
PaintShop Pro, Bildbearbeitung 387
Paketmanager, Linux 204
Paketvermittlung 538
Panel, AWT-Klasse 304, 311
Pantone, Sonderfarbskala 390
papd, Netatalk-Dienst 651
Papierformat 433
DIN 433
Nicht-DIN 434
Tageszeitungen 434
Papierkorb, MacOS X 210
Parallele Datenübertragung 115
Parallelport 122
Parameter
bei Systemprogrammen 175
von Perl-Subroutinen 274
Parameter-RAM (Mac) 115
Parameter-RAM, löschen 115
PARC, Xerox-Forschungszentrum 150
Parent-Prozess 159, 998
Parity Bit → Prüfbit
parseFloat(), JavaScript-Methode 860
parseInt(), Java-Methode 279
parseInt(), JavaScript-Methode 859
Partitionierung 998
praktische Durchführung 127
Partitionierung (Festplatten) 126
Pascal, Blaise 34
Pascal, Programmiersprache 46
passwd, UNIX-Befehl 188
Passwort, ändern, UNIX 188
Passwörter 986
Brute-Force-Attacke 173
crack (Knackprogramm) 172
Erzeugungstipps 173
patch, UNIX-Befehl 185
Patches, Sicherheit 975
path, Umgebungsvariable 175
Windows 221
pause(), UNIX-Systemaufruf 159
PC 40
Aufbau 97
Geschichte 149
Zentraleinheit 98
PC-Card → PCMCIA-Anschluss
PCDATA, Text in XML 692
PCI 119, 998
PCMCIA-Anschluss 119
PC-Netzwerke, Entwicklung 543
PDA (Personal Digital Assistent) 42
PDF 447
erstellen 448
InDesign 447
PDF-X 449, 998
PDP, Kleincomputerserie von DEC 148
PDP, Kleinrechnerserie von DEC 38, 46
PEAR, PHP-Erweiterungen 811
Pentium, Prozessorfamilie 100, 105
Peripherie
Ausgabegeräte 136
Digitalkamera 136
Drucker 140
Eingabegeräte 134
Einsteckkarten 119
Grafikkarte 137
Maus 134
Monitor 138
Scanner 135
Tastatur 134
Übersicht 123
Perl 998
Perl, Programmiersprache 264
!~-Operator 294
$ 266, 267
$/-Variable 517
$_ 269
$-Operator, RegExp 293
% 268
&&-Operator 273
**-Operator, RegExp* 291
+-Operator, RegExp 291
.-Operator 272
=~ 276
=~-Operator 294
?-Operator 291
@ 267
@_ 274
^-Operator, RegExp 291, 293
|| zur Fallentscheidung 272
||-Operator 272
|-Operator, RegExp 293
accept()-Funktion 669
ActivePerl-Distribution 265

alternative Zeichen in RegExp 291
Anführungszeichen 271
Array-Elementanzahl 280
Arrays 267
Arrays zusammenfassen 269
Ausdrücke 270
Ausgabe 266
Autoflush-Modus 668
Backticks 297
bind()-Funktion 669
chomp-Funktion 266
close 275
closedir 275
connect()-Funktion 668
Dateien bearbeiten 274
Dateihandles 275
defined 273
Dereferenzierung 270
die 273
Eingabe 266
elsif 273
eq-Operator 272
Ersetzen durch RegExp 294
foreach-Schleife 269
ge-Operator 272
gethostbyname()-Funktion 666
getprotobyname()-Funktion 665
getservbyname()-Funktion 667
gt-Operator 272
Hashes 268
HIER-Dokumente 272
if, nachgestelltes 273
IO 668, 669
join 269
keys 270
Kommentare 271
le-Operator 272
Listen 268
listen()-Funktion 669
Listenfunktionen 268
Literale 270
lt-Operator 272
m//-Operator 291
MacPerl 265
my 267
ne-Operator 272
Objektorientierung 670
open 275
opendir 275
-Operator 266
Operatoren 270
Parameter von Subroutinen 274
pipe() 299
Pipes verwenden 298
pop 268
print-Anweisung 266
Programmabbruch 273
Programme starten 265
push 268
q// 271
qq// 271
Queue 286
qw// 271
readdir 275
Referenzen 270
Reguläre Ausdrücke 276, 290
s///-Operator 276, 291, 294
Shebang 265
shift 268
Skalare 266, 267
Socket-Modul 665
split 269
Stack 286
strenge Wertüberprüfung 267
strict 267
Strings zerlegen 269
String-Vergleiche 271
String-Verkettung 272
sub 273
Subroutinen 273
tr///-Operator 295
unless 273
unshift 268
use-Direktive 267
Variablen 267
Variablenexistenz prüfen 273
Variablensubstitution 266
Verzeichnisse verarbeiten 275
-w, Modifikator 265
Warnungen aktivieren 265
Zeichengruppen in RegExp 291
Zeilenumbruch 517
Zeilenumbrüche entfernen 266
Perl, Programmiersprachen//-Operator 294
Perl-Anweisung 273
Permutationen 83
Personal Computer → PC
Personal Digital Assistant (PDA) 42
Petabyte 70
Pfad
absoluter 165, 168
in HTML-Hyperlinks 735
relativer 165, 168
unter UNIX 165
unter Windows 167
Pfade 412
ändern 414
Ankerpunkte 412
Effekte, Illustrator 417
Flash 919
Illustrator 412
InDesign 441
Photoshop 411
trainieren 414
Zeichenstift, Illustrator 413
PGP 988
Phishing 979
Photocaster, Director-Xtra 492
PhotoImpact, Bildbearbeitung 387
Photoshop 387
Alphakanal 402
Anti-Alias 400
Arbeitsflächengröße 397
Arbeitsoberfläche 394
Auswahl 399
Beschnittgruppen 410
Bildauflösung 395
Bildgröße 395
Bildgrößeninterpolation 396
Buntstift 404
Deckkraft 406
Ebenen 407
Ebenenmaske 409

Ebenenoptionen 408
Einstellungsebenen 410
Farbbalance 404
Farbkanäle 401
Farbkorrektur 403
Farbseparation, Einstellungen 399
Farbton 404
Farbverlauf 406
Filter 411
Füllebenen 410
Füllwerkzeug 406
Glätten 400
Hintergrundebene 407
Histogramm 404
Kopierstempel 406
Lasso 400
Lupe 397
Malmodus 405
Malwerkzeuge 404
Maskierungsmodus 403
Pfade 411
Pinsel 404
Radiergummi 407
Sättigung 404
Text 411
Tonwertkorrektur 403
Transformation 403
Zauberstab 400
Photoshop Elements 387
PHP 810, 999
$, Variablenbeginn 818
Anführungszeichen 820
Arrays 819
Arrays als Hashes 820
Arrays, mehrdimensionale 821
Bezeichner 818
Call by Reference 824
Cookies 834
Datei-Uploads 832
Datentyp testen 826
Datum und Uhrzeit 839
Diskussionsforum 851
Dokumentaufbau 810
Dynamische Variablen 819
each()-Funktion 821
echo()-Anweisung 817
erste Beispiele 817
explode()-Funktion 821
extends, Schlüsselwort 829
fclose()-Funktion 836
feof()-Funktion 836
fgets()-Funktion 836
fopen()-Funktion 835
fputs()-Funktion 838
function, Schlüsselwort 823
Funktionen 823
Funktionsparameter 824
Funktionswertrückgabe 825
Gästebuch, einfaches 837
Gästebuch, vollständiges 842
global 824
Globale Variablen 824
HTML ausgeben 817
implode()-Funktion 821
include()-Funktion 830
Include-Dateien 830
Installation 811
Installation, UNIX 811
Installation, Windows 813
is_float()-Funktion 826
is_int()-Funktion 826
is_null()-Funktion 826
is_string()-Funktion 826
Kommentare 823
Konstruktor 828
Mehrdimensionale Arrays 821
mysqli-Schnittstelle 849
Objektorientierung 826
Operatoren 819
PEAR-Erweiterungen 811
php.ini, Konfigurationsdatei 814
phpMyAdmin 368
preg_match()-Funktion 822
preg_replace()-Funktion 822
Quizspiel (MySQL-Anwendung) 850
Referenzen 824
Reguläre Ausdrücke 822
return-Anweisung 825
Sessions 833
sizeof()-Funktion 820
Sprachgrundlagen 818
Strings zerlegen 821
Textdateien verarbeiten 835
time()-Funktion 839
Variablen 818
Variablen, dynamische 819
Vererbung 829
Webspezifische Funktionen 830
XAMPP, integrierter Server 652
PHP, date()-Funktion 839
php.ini, Konfigurationsdatei 814
phpMyAdmin 368
Pica-Point, Schriftgröße 422
PICT, Bilddateiformat 533
PID (Prozess-ID) 158
Pigmentfarben 389
ping, TCP/IP-Dienstprogramm 640
Ergebnisse auswerten 641
Windows 641
Pipe 999
pipe(), Perl-Funktion 299
Pipeline (CPU-Warteschlange) 104
Pipes
Anwendung 298
in Programmiersprachen 296
in UNIX-Shells 178
Perl 298
Windows 221
zur Inter-Prozess-Kommunikation 160
Pixelgrafik 50, 387
Pixelwiederholung, Bildgrößeninterpolation 396
Planung, Software-Engineering 327
Platzhalter
in UNIX-Dateinamen 179
Windows-Dateinamen 221
Plug & Play 119, 999
PNG, Bilddateiformat 532, 780
in Fireworks 784
Polymorphe Viren 973
Polynomielle Komplexität 84

pop, Perl-Anweisung 268
POP3 623, 999
Befehle 624
Sitzung 623
Ports, TCP 603
Ports, UDP 605
position, CSS-Angabe 769
POSIX 999
POSIX-Standard 148
POST 111
POST (BIOS-Selbsttest) 111
POST, HTTP-Befehl 628
zum HTML-Formularversand 750
Post-Dekrement 238
PostgreSQL 365
Post-Inkrement 238
PostScript 141, 526, 999
Datei erzeugen 448
EPS 526
Farbe 448
PPD 526
PostScript-Schrift 440
Power Management 113
Power-On Self Test → POST
PowerPC, Prozessor 103
PPD, Druckerbeschreibungsdatei 526
PPP 999
PPP-Protokoll, DFÜ 570
Prä-Dekrement 237
Prädikatenlogik 48, 53, 999
Präemptives Multitasking 154
Prä-Inkrement 237
Praktische Informatik 26
Präprozessor, C 250
#*define* 251
#*endif* 251
#*ifdef* 251
#*include* 230, 250
Präsentation, OSI-Schicht 545
Preflight 999
Preflight, DTP-Test 438
preg_match(), PHP-Funktion 822
preg_replace(), PHP-Funktion 822
Premiere
A/B-Bearbeitung 462
Arbeitsablauf 463
Arbeitsoberfläche 461
Audio-Volumen 462
Clips bewegen 465
Filmdatei exportieren 466
Projektfenster 462
Projektvorgaben 461
Schnittfenster 462
Titel erzeugen 465
Transparenz 465
Überblendungen 465
Video-Deckkraft 462
Werkzeugleiste 463
Primärfarben 389
Primärschlüssel 359, 999
Primärschlüssel, einrichten, SQL 377
PRIMARY KEY, SQL-Feldoption 377
print(), Java-Methode 255
print, Perl-Anweisung 266
printf(), C-Funktion 230, 248
Formatangaben 231
println(), Java-Methode 255
Prisma 388
private, Java-Kapselung 258
Problemorientierte Programmiersprachen 44
Programmablaufplan 81
Programmgesteuerter Rechenautomat 32
Programmiersprachen 43
Ada 35
Algorithmen 277
Anweisungsblock 239
Assembler 43
BASIC 45
Baum, Datenstruktur 288
Binäre Suche 285
C 46, 148, 228
C# 48
C++ 48
Cobol 45
Compiler 44, 107
Datenstrukturen 277, 285
deklarative 48
Fortran 45
Funktion 46
funktionale 48
GUI-Programmierung 303
Imperative 46
Interpreter 45, 264
Iteration 282
Java 48, 251
Kontrollstrukturen, C 239
Lineare Suche 284
Lisp 48, 198
logische 48
Logo 48
Maschinensprache 43
mit Datenbanken arbeiten 382
Modularisierung 46, 243
Objective-C 207
objektorientierte 47
Objektorientierung 252
Pascal 46
Perl 264
PHP 810
Pipes 296
problemorientierte 44
Prolog 48
Prozedur 46
Prozedurale 46
Rekursion 281
Skriptsprachen 45, 264
SmallTalk 48
Sortier-Algorithmen 279
Strukturierung 46, 243
Such-Algorithmen 284
Systemaufrufe 108, 295
Threads 299
Turing-Vollständigkeit 87
Unicode-Unterstützung 522
Programmstrukturierung 46
Projektmanagement 328
Netzplan 329
Prolog, Programmiersprache 48
PROM 110
prompt(), JavaScript-Methode 858
Prompt, UNIX 173
Proof 439, 999

Proportionalschrift 425, 999
protected, Java-Kapselung 260
Prozedur 46
Prozedurale Programmiersprachen 46
Prozess 999
Management durch Betriebssystem 145
Prozessanalyse 331
Prozesse
als Baum anzeigen 189
Benutzermodus 158
Child-Prozess 159
Deadlock 160
Definition 158
duplizierte 296
im Benutzermodus 153
im Kernelmodus 153
in den Hintergrund stellen 175
init 158
Kernelmodus 158
Kommunikation 160
Multitasking 158
Parent-Prozess 159
Prozess-ID (PID) 158
Race Condition 160
Signalverarbeitung 159
Threads als Alternative 161
unter UNIX 158
Unterstützung durch CPU 101, 106
Verwaltung durch Betriebssysteme 158
Verwaltung, UNIX 189
Windows 160
Prozessfarben 390
Prozess-ID 158
Prozessmanagement 145
Prozessor 36, 98, 102
3D Now! 105
Adressbus-Wortbreite 102
Alpha 103, 105
als Bauteil 100
ALU 100
AMD 100
Arbeitsweise 105
Architektur 104
Athlon 100, 105
Aufbau 99
Bedingter Sprung 106
Befehlstabelle 101
Befehlszeiger 100
Bestandteile 100
Bus 101
Cache 101
CISC 104
Datenbus-Wortbreite 102
der Grafikkarte 137
Effizienz 104
FLOPS 104
Intel 100
Itanium 103
Maschinenbefehle 107
MIPS 104, 105
MMX 105
MosTek 6502 103
Motorola 68000-Familie 103
Pentium-Familie 100, 105
Pipeline 104
PowerPC 103
Prozesse 106
Register 100
Registerwortbreite 102
RISC 104
Sprungbefehle 106
Stack 106
Steuerbus-Wortbreite 102
Steuerwerk 100
Sun SPARC 105
Taktfrequenz 103
Übertakten 103
Unbedingter Sprung 106
Unterprogramm-Aufrufe 106
Wortbreite 102
Wortbreiten-Vergleich 103
Z80 103
Prozessor, virtueller 87
Prozessorarchitektur 104
Prozessverwaltung 158
Prüfbit 116
Prüfung
IT-Berufe 28
Mediengestalter 30
ps, UNIX-Befehl 159, 189
PS/2-Anschluss 122
PSD, Bilddateiformat 530
Pseudocode, zur Algorithmendarstellung 81
pstree, UNIX-Befehl 189
PTR-Record (DNS) 613
public, Java-Kapselung 254, 258
Public-Key-Verschlüsselung 987
Pulswahlverfahren 571
Punkt, Schriftgröße 422
push, Perl-Anweisung 268
pwd, UNIX-Befehl 181

Q

q//, Perl-Quoting 271
qq//, Perl-Quoting 271
Qt, Grafikbibliothek 199
Quadratische Komplexität 84
Qualitätsfaktor, Scan-Auflösung 394
Qualitätsmanagement 329
Quantencomputer 42
QuarkXPress 437
Quartz, Grafikbibliothek 207
Qubit 42
Queue, Datenstruktur 285
Perl 286
QuickSort 999
QuickSort, Algorithmus 282
Funktionsprinzip 282
Median 282
QuickTime, Multimedia-Technologie 207
QuickTime, Video-Dateiformat 535
qw//, Perl-Quoting 271

R

Race Condition 160, 999
Radiergummi, Photoshop 407
Rahmen, Gestaltungselement 436
RAID 127, 999
Levels 127
Mirroring 127
Stripe Set 127
Stripe Set mit Parity 127

RAM 98, 108
als Bauteil 108
Auslagerungsdatei 162
Bedeutung in der Speicherhierarchie 102
CMOS 111
DDR-RAM 109
der Grafikkarte 137
DIMM-Module 109
Dynamic 108
EDO 109
einbauen 109
empfohlene Menge 109
FP 109
Paging 162
Rambus 109
RD-RAM 109
RIMM-Module 109
SD-RAM 109
Segmentierung 162
Seitenfehler 162
SIMM-Module 109
Static 108
Verwaltung durch das Betriebssystem 161
virtuelle Adressierung 161
Rambus-RAM 109
RAMDAC (Grafikkarte) 138
Random Access Memory → RAM
Rangfolge, durch Klammern ändern 238
Rangfolge, Operatoren 238
Rasterweite 392
Rasterwinkel 392
Rational Unified Process 336
Rationale Zahlen 62
Raytracing 999
Raytracing, 3-D-Rendering 474
rd → rmdir, Windows-Befehl
RDBMS 358, 999
RD-RAM 109
readdir, Perl-Anweisung 275
readLine(), Java-Methode 255
Read-only Memory → ROM
REAL, SQL-Datentyp 375
Rechenautomat 32
Rechenbefehle, des virtuellen Prozessors 89
Rechendaten 355
Rechenmaschinen, mechanische 34
Rechentafel 34
Rechenwerk 97
Rechteckauswahl, Photoshop 399
Record (Datensatz) 357
Red Book (Audio-CD) 129
Red, Green, Refactor (TDD) 351
Reelle Zahlen 62
Refactoring 322
Referenzen, auflösen, Perl 270
Referenzen, Perl 270
Reflektion, Farbe 388
regedit, Windows-Dienstprogramm 225
RegExp → Reguläre Ausdrücke
Regionale Einstellungen, MacOS X 212
Register 999
der CPU 100
des virtuellen Prozessors 88
Wortbreite 102
Register (Schaltung) 79
Registermaschine 87, 999
Registrierdatenbank → Registry, Windows
Registry, Windows 225
Reguläre Ausdrücke 290, 999
!~, Operator 294
=~, Perl-Operator 294
Alternative Textteile 293
Alternative Zeichen 291
beliebig viele Zeichen 291
ein oder mehr Zeichen 291
ersetzen 294
Escape-Sequenzen 292
grep 184
Groß- und Kleinschreibung ignorieren 293
in PHP 822
JavaScript 870
Klammern 293
Leerzeichen 292
m//, Operator 294
m//-Operator 291
Matching 294
mehrere Treffer finden 293
Modifikatoren 293
Muster 291
optionale Zeichen 291
Perl 276, 290
s///, Operator 294
s///-Operator 291
Sonderzeichen 292
Teilausdrücke 293
Whitespace 292
Wortgrenzen 293
Wortzeichen 292
Zeichen ausschließen 291
Zeichenanzahl 292
Zeichengruppen 291
Zeilenanfang 293
Zeilenende 293
Ziffern 292
reiserfs, Linux-Dateisystem 186
Rekursion 281, 999
Relais 35
Relationale Datenbak, Join-Abhängigkeiten 364
Relationale Datenbank 356, 358, 999
1:1-Relation 359
1:n-Relation 359
Access 365
Änderungsabfrage 373
Arten 365
atomare Information 362
Auswahlabfrage 361, 373
Boyce-Codd-Normalform 363
Desktop-Datenbank 365
Einfügeabfrage 373
FileMaker 365
freie Server 365
Fremdschlüssel 359
Grenzen 365
Index 359
Java-Programmierung 382
JDBC 382
kommerzielle Server 365

Konsistenz 358
Löschabfrage 373
m:n-Relation 359
MySQL 368
Normalformen 362
Normalisierung 362
ODBC 382
PostgreSQL 365
Primärschlüssel 359
Primärschlüssel einrichten, SQL 377
Programmierung 382
Relationen 359
Schlüssel 359
SQL 362
Tabelle erzeugen, SQL 373
Tabelle löschen, SQL 374
Relationale Datenbanken, SQL 372
Relationen
im RDBMS 359
Relativer Pfad 165, 168
Remote Method Invocation → RMI, Java
Renaissance-Antiqua
Französische 426
Venezianische 426
rename, Windows-Befehl 222
repaint(), AWT-Methode 309
repeat, Lingo-Schleife 500
Replikation 539
Request For Comments → RFC
Resource Fork, HFS 213
Creator ID 213
File Type ID 213
Ressourcen
Hardware- 117
Plug & Play 119
Zuweisung 119
Ressourcenmanagement 329
Restmenge 64
ResultSet, JDBC-Klasse 384
return, C-Anweisung 231, 243
RFC 540, 999
1300 540
2131 und 2123, DHCP 595
1034 und 1035, DNS 607
959, FTP 617
2324, HTCPTP 540
2616, HTTP 627, 652
2060, IMAP 625
791, IP-Protokoll 577
2460, IPv6 597
2045 bis 2049, MIME 621
977, NNTP 625
2178, OSPF 594
1918, private IP-Adressen 581
1723, RIP-2 594
821, SMTP 619
2821, SMTP 619
3330, Spezial-IP-Adressen 581
793, TCP 601
854, Telnet 616
822, Textnachricht 621
2822, Textnachricht 621
768, UDP 601
1738, URL 627
RGB, Photoshop-Farbmodus 398
RGB-Farbe 50, 389
Rhapsody (Mac OS X) 206
RIMM-Module (RAM) 109
Ring-Topologie, Netzwerk 552
RIP 1000
RIP, Routing-Protokoll 593
RISC 1000
RISC-Prozessor 104
Beispiele 105
Risikomanagement 329
Ritchie, Dennis 46, 148, 228
rm, UNIX-Befehl 181
rmdir, UNIX-Befehl 182
rmdir, Windows-Befehl 222
RMI (Remote Method Invocation) 1000
RMI, Java 676
robots.txt, Suchmaschinen-Info 761
Röhrenmonitor 138
Röhrenrechner 36
Rollback (Transaktionen) 364
Rollover-Effekt
Dreamweaver 795
in Fireworks 787
JavaScript 878
praktische JavaScript-Lösung 879
ROM 98, 110
Bauarten 110
Bedeutung 99
bei 8-Bit-Homecomputern 99
BIOS 110
Römische Zahlen 65
root, Benutzer 159, 172
Home-Verzeichnis 165
temporär arbeiten als 177
root, UNIX-Verzeichnis 165
root, unter Mac OS X 212
Rootkits 983
ROT13 986
Roter Balken (Unit-Test) 350
Rouitng-Tabellen 591
Round-Robin-DNS 614
route, UNIX-Befehl 637
Router 1000
Router, IP-Protokoll 589
Routing
Autonomes System 592
IP-Protokoll 589
Routing-Protokolle 592
BGP 594
OSPF 594
RIP 593
Routing-Tabellen, anzeigen 591
rpm, Linux-Paketmanager 204
RS-232 122
RS-Flip-Flop 78
Rumbaugh, James 339
run(), Java-Methode 300
Runlevel 203
Runnable, Interface 300
Runnable, Java-Interface 262

S

s///, Perl-Operator 276, 291, 294
Samba 649, 1000
als Client für Windows-Server 650
Drucker freigeben 650
globale Parameter 650

Konfiguration 649
smb.conf, Konfigurationsdatei 649
starten 649
Verzeichnisse freigeben 650
Windows-Freigabeart 650
Sampling 1000
Sampling, Audio 50
Sampling, Audio-Digitalisierung 453
Sampling-Rate 51
Sampling-Tiefe, Audio 51
Satellit, DSL-Verbindung 574
Satz, Umfließen 436
Satzspiegel 430
SAX 1000
Beispielprogramm 715
ContentHandler, Interface 714
ContentHandler-Callbacks 714
ContentHandler-Methoden 714
Dokument parsen 713
DTDHandler, Interface 714
EntityResoler, Interface 714
ErrorHandler, Interface 714
Event-Handler 714
InputSource, Klasse 713
Parser-Instanz erzeugen 713
XMLReader-Interface 713
SAX (Simple API for XML) 713
SaX (X-Server-Konfigurationsprogramm) 199
sbin, UNIX-Verzeichnis 164
scanf(), C-Funktion 241, 248
Scanner 135
Auflicht 135
Bildauflösung 394
Diascanner 136
Durchlicht 135
Flachbettscanner 135
Kleinbildscanner 136
Qualitätsfaktor 394
Trommelscanner 135
Schall 453
Schallwellen 453
Schaltalgebra 26, 73
Schaltkreisvermittlung 538
Schaltung, Register 79
Schichtenmodell 1000
Alltagsbeispiel 547
Mail-Beispiel 549
Praxis 547
Schleifen
C 242
do/while() 242
Endlosschleifen 283
for() 243
foreach, Perl 269
fußgesteuerte 242
in Shell-Skripten 191
kopfgesteuerte 242
Lingo 500
mit break abbrechen 283
Schleifenrumpf 242
Schleifenzähler, Variable 242
Schlüssel
eines Hashes 269
im RDBMS 359
Schlussfolgerung, logische 55
Schlussfolgerung, Umkehrschluss 56
Schmuckfarben 390
Schnittmenge 63
Schnittstellen, Hardware 99
Schnittstellen, Softwareentwurf 334
Schreib-Lese-Kopf, der Turing-Maschine 85
Schreibschriften 428
Schriftart, in HTML einstellen 729
Schriftarten, im Drucker 142
Schriften
Bitmap 440
Dateien 439
OpenType 440
PostScript 440
TrueType 440
Schriftgröße
Dickte 423
Didot-Punkt 422
DTP-Punkt 422
in HTML 728
Kegelhöhe 422
Mittellänge 423
Oberlänge 423
Pica-Point 422
Punkt 422
traditionelle Namen 422
Unterlänge 423
Versalhöhe 423
Schusterjunge, Absatzfehler 433
screen, JavaScript-Objekt 893
Scribble, Layout-Planung 437
SCSI 121, 1000
anschließen 121
ID 121
Terminator 121
SCSI-ID 121
SDI, Windows-Anwendungen 218
SD-RAM 109
SDSL 573
sed 264, 290
Sedezimalsystem
→ Hexadezimalsystem
Segmentierung (Speicher) 162
Seiten (Speicher) 162
Seitenfehler (Speicher) 162
Seitentabelle (Speicher) 162
SELECT, SQL-Abfrage 377
Semikolon, Abschluss von Anweisungen 230
Sequenzdiagramme (UML) 343
Serialisierung 262
Serializable, Java-Interface 262
Serielle Datenübertragung 115
Bedeutung 116
Kontrollbit 116
Leitungskonventionen 116
Prüfbit 116
Startbit 116
Stoppbit 116
Serifenbetonte Linear-Antiqua 428
Serifenlose Linear-Antiqua 428
Server, Netzwerk 553
Server, Windows-Betriebssysteme 214

Serverdienste
Anwendungsserver 557
Application Server 557
Dateiserver 554
Druckserver 555
Einrichten, Linux 647
Einrichten, MacOS X 644
Einrichten, UNIX 647
Einrichten, Windows 644
File-Server 554
HTTP-Server 556
Mail-Server 555
netatalk 651
Print-Server 555
Samba 649
Übersicht 554
Webserver 556
Server-Gefahren 982
Serversysteme, Macintosh 643
Servlets, Java 680
Session-Hijacking 983
SET, SQL-Befehl 381
SET, SQL-Datentyp 376
setColor(), AWT-Methode 307
setLayout(), AWT-Methode 312
Setter-Methoden, Java 322
setTimeout(), JavaScript-Methode 875
SGML 1000
HTML-DTD 719
XML als moderne Version 685
sh (Bourne-Shell) 174
Shannon-Theorem 453
Share Level Security, Samba 650
Share Level Security, Windows-Freigabeart 645
Shebang
bei Shell-Skripten 190
Perl 265
Shell 157
Ausgabeumleitung 177
bash 174
Befehle, Windows 222
Bourne-Shell 174
Cmd.exe, WinNT 221
COMANND.COM, MS-DOS 221
C-Shell 174
Eingabeaufforderung 173
Eingabeumleitung 178
Eingabevervollständigung 176
ermitteln, welche läuft 174
Escape-Sequenzen 185
HIER-Dokument 178
History 176
Korn Shell 174
Pipes 178
Prozesse in den Hintergrund stellen 175
Shell-Skripte 190
UNIX 157
unter MacOS X 208
Windows 221
shell, Befehle als root ausführen 177
Shell-Skripte 190
Beispiel 192
case-Befehl 191
Fallentscheidungen 191
for-Befehl 191
if-Befehl 191
Schleifen 191
Shebang 190
Variablen 191
while-Befehl 191
shift, Perl-Anweisung 268
short, C-Datentyp 233
Short-Circuit-Logik 273
show(), AWT-Methode 305
shutdown, UNIX-Befehl 190
Sicherheit 971
Admin-Rechte 976
AdWare 979
Backdoors 977
Backups 975
CGI u. Webanwendungen 984
chroot-Umgebung 985
Crackerangriffe 983
Crackertools 985
Cross-Site-Scripting (XSS) 983
Dialer 978
Exploits 982
Firewall 975
Firewalls 985
Hoaxes 982
Intrusion Detection Systems 985
keine absolute 971
Kettenmails 982
Kryptografie 986
Man-in-the-Middle-Angriff 983
Menschliches Versagen 986
MySQL (UNIX) 369
MySQL (Windows) 371
Passwörter 986
Patches installieren 975
PC-Gefahren 972
Phishing 979
Rootkits 983
Server-Gefahren 982
Session-Highjacking 983
Social Engineering 986
Spam 980
SpyWare 979
SQL-Injection 984
Viren 972
Würmer 973
Sicherung, OSI-Schicht 544
SIGALRM, Signal 159
SIGHUP, Signal 159
SIGINT, Signal 190
SIGKILL, Signal 159
Signal 1000
Signale
an Prozesse senden 190
SIGALRM 159
SIGHUP 159
SIGINT 190
SIGKILL 159
SIGTERM 159
Verarbeitung durch Prozesse 159
zur Inter-Prozess-Kommunikation 160
Signatur, digitale 988
signed, C-Datentyp 233
SIGTERM, Signal 159

Silbentrennung, für Blocksatz 432
Silicon Valley 37
Silizium 37
SIMM-Module (RAM) 109
Simulation, eines Prozessors 87
Sinclair ZX Spectrum 149
Sinclair ZX81 40, 149
Sinclair, Clive 40
Single Document Interface → SDI, Windows-Anwendungen
Singleton, Entwurfsmuster 346
Implementierung (Java) 348
Sinuskurve, zeichnen, AWT 307
Sitzung, OSI-Schicht 544
sizeof(), PHP-Funktion 820
Skalare, Perl-Variablen 266, 267
Skriptsprachen 45, 264
Slash, Blog-Tool 809
Slave-Nameserver 609
Slots, Prozessoren 100
SMALLINT, SQL-Datentyp 375
Smalltalk 325
SmallTalk, Programmiersprache 48
smbclient, Samba-Dienst 650
SMB-Protokoll, Windows-Netzwerk 633
SMPTE-Zeitformat, für Video 462
SMTP 619
Befehle 620
Sitzung 620
Snort 985
SOAP 1000
Clients 683
SOAP, Web Services 682
SOA-Record (DNS) 613
Social Engineering 986
sockaddr_in, Netzwerkprogrammierung 666
Sockel, für Prozessoren 100
Socket 1000
Socket, Perl-Modul 665
Sockets 664
accept(), Funktion 669
Adressen 666
bind(), Funktion 669
connect(), Funktion 668
Datagramme senden und empfangen 667
Domain 665
erzeugen 665
IO 669
IP-Adresse 666
lauschende 669
listen(), Funktion 669
Protokoll 665
TCP 667
TCP-Client 668
TCP-Port 666
TCP-Server 669
Typ 665
UDP 667
Verbindung aufnehmen 669
Software installieren, Windows 223
Software-Engineering 325
Agile Entwicklungsprozesse 338
Analyse 330
CASE-Tools 340
Code-Review 335
Dokumentation 335
Entwicklungsprozesse 336
Entwicklungszyklus 326
Entwurfsmuster 345
Entwurfsphase 333
Extreme Programming 338
Implementierungsphase 334
in der IT-Ausbildung 326
Lastenheft 331
mit Eclipse 318
Objektorientierte Analyse 331
Objektorientierung 325
Pflichtenheft 332
Planungsphase 327
Projektmanagement 328
Projektphasen 326
Refactoring 322
Schnittstellen 334
Spiralmodell 327
Strukturierte Analyse 331
Test-first-Verfahren 338, 351
Testphase 335
UML 339
Unified Process 336
Unit-Tests 335, 348
Wasserfallmodell 327
Softwareentwicklung → Software-Engineering
Softwarekrise 325
Softwaretechnik → Software-Engineering
Solaris 1000
Solaris, Betriebssystem 148
Sonderfarben 390
HKS-Skala 390
Pantone-Skala 390
Sortier-Algorithmen 279
BubbleSort 279
QuickSort 282
Soundeffekte 456
Soundkarte 142
Anschlüsse 142
Audio-CDs abspielen 142
MIDI 142
SP-DIF-Anschlüsse 142
Source, Stromeingang des Transistors 74
Spam 980
SpamAssassin 981
SP-DIF-Anschlüsse 142
Special Files → Gerätedateien
Spectrum, Homecomputer 149
Speicher
Management durch Betriebssystem 145
RAM 98
ROM 98
virtueller 102
Speicher reservieren, C 287
Speicheradressierung 69
Speichermanagement 145
Speicherseiten 162
Speicherverwaltung 161
x86-Systeme 162

Speicherzelle (Schaltung) 78
Spektrum 388
Spielautomat, Flash-Anwendung 959
Spiralmodell 327
splendid, Typographie 424
Spline 1000
split, Perl-Anweisung 269
Sprache (Umgangssprache), zur Algorithmendarstellung 81
Sprites, Director 479
Sprungbefehle
bedingte 106
beim virtuellen Prozessor 90
der CPU 106
unbedingte 106
Sprungvorhersage 102
Sprungvorhersage → Prozessor
SpyWare 979
SQL 362, 372, 1000
Aggregatfunktionen 379
Änderungsabfrage 373, 381
Arithmetische Operatoren 378
AS-Klausel 378
Auswahlabfrage 373, 377
AUTO_INCREMENT, Feldoption 377
BIGINT, Datentyp 375
BINARY, Feldoption 376
BLOB, Datentyp 376
CHAR, Datentyp 376
COUNT-Funktion 379
CREATE DATABASE, Befehl 373
CREATE TABLE, Befehl 373
DATE, Datentyp 375
Datentypen 375
Datentypen in Java 384
Datentypen, Aufzählung 376
Datentypen, Binärobjekte 376
Datentypen, Datum und Uhrzeit 375
Datentypen, Fließkomma 375
Datentypen, ganzzahlige 375
Datentypen, Text 375
DATETIME, Datentyp 375
DECIMAL, Datentyp 375
DEFAULT, Feldoption 377
DELETE-Abfrage 381
DOUBLE, Datentyp 375
DROP DATABASE, Befehl 374
DROP TABLE, Befehl 374
Einfügeabfrage 373, 380
ENUM, Datentyp 376
Felder mit Nullen füllen 376
Feldoptionen 376
Feldwert einmalig machen 377
FLOAT, Datentyp 375
Funktionen 379
Indizes erstellen 377
Inner Join 379
Inner Join durch WHERE ausdrücken 380
INNER JOIN-Klausel 380
INSERT-Abfrage 380
INT, Datentyp 375
Joins 379
LIKE-Klausel 378
LONGBLOB, Datentyp 376
LONGTEXT, Datentyp 376
Löschabfrage 373, 381
MAX-Funktion 379
MDEIUMBLOB, Datentyp 376
MEDIUMTEXT, Datentyp 376
MIN-Funktion 379
Mustervergleich 378
NULL, Feldoption 377
ORDER BY-Klausel 379
Primärschlüssel einrichten 377
PRIMARY KEY, Feldoption 377
REAL, Datentyp 375
SELECT-Abfrage 377
SET, Datentyp 376
SET-Befehl 381
SMALLINT, Datentyp 375
Sortieren 379
Standardwert angeben 377
SUM-Funktion 379
TEXT, Datentyp 376
TIME, Datentyp 375
TIMESTAMP, Datentyp 375
TINYBLOB, Datentyp 376
TINYINT, Datentyp 375
TINYTEXT, Datentyp 376
UNSIGNED, Feldoption 376
UPDATE-Abfrage 381
VARCHAR, Datentyp 376
Vergleichsoperatoren 378
Volltextindex 377
WHERE-Klausel 378
YEAR, Datentyp 375
ZEROFILL, Feldoption 376
SQL-Injection 984
SQL-Server, Microsoft 365
SRAM 108
SSH 988, 1000
SSH (Secure Shell) 616
SSL 988
Stack
der CPU 101, 106
des virtuellen Prozessors 91
Java-Klasse 264
Stack Pointer → Stack-Zeiger
Stack, Datenstruktur 285
Perl 286
Stack-Overflow 106, 1000
Stack-Zeiger 101, 106
Stallman, Richard 170
Stammdaten 355
Standardausgabe (stdout) 177
Standardausgabe, Java 255
Standardeingabe (stdin) 177
Standardfehlerausgabe 303
Standard-Fehlerausgabe (stderr) 177
Stapelverarbeitung 147
Startbit 116
Startvolume, Mac OS X-Einstellung 211
startx, X Window starten 199
Statement, JDBC-Klasse 383
Static RAM → SRAM
static, Java-Methoden 254
static, Variableneigenschaft, C 234
Statische Variablen 234

stddef.h, C-Bibliothek 234
stderr 303
stderr, Standard-Fehlerausgabe 177
stdin, Standardeingabe 177
stdio.h, C-Bibliothek 230, 248
stdout, Standardausgabe 177
Stealth-Viren 973
Stellenwertsystem 34, 64
Basis 65
Stern-Topologie, Netzwerk 552
Steuerbus 101
Wortbreite 102
Steuerwerk 96
der CPU 100
Stoppbit 116
strcat(), C-Funktion 249
strcmp(), C-Funktion 249
strcpy(), C-Funktion 249
Stream-Sockets 665
strftime(), C-Funktion 250
strict, Perl 267
String 230, 1000
aus Datei lesen, C 249
Eingabe, C 248
String, Java-Datentyp 255
string.h, C-Bibliothek 249
String-Literale 235
Strings
aufteilen, Java 257
aus Arrays, Perl 269
Darstellung in C 246
Darstellung, C 230
einlesen, C 231
einzelne Zeichen lesen, Java 257
Funktionen in C 249
in GUI schreiben 308
Java 255
Kopieren, C 249
Länge ermitteln, Java 257
Null-terminierte 246
Position ermitteln, Java 257
vergleichen, C 249
vergleichen, JavaScript 860
vergleichen, Perl 271
Verketten, C 249
Verkettung, Java 255, 256
Verkettung, Perl 272
zerlegen, Perl 269
zerlegen, PHP 821
Strings, Operationen in Java 256
String-Verkettung, Perl 272
Stripe Set (RAID) 127
Stripe Set mit Parity (RAID) 127
Stroustrup, Bjarne 48
struct, C 247
Structural Patterns → Strukturmuster
Structured Query Language → SQL
Strukturen, C 247
Strukturierte Analyse 331
Strukturierung von Programmen 46
Strukturierung, Programme 46, 243
Strukturmuster 345
Studiengänge, Informatik 31
Style Sheets → CSS
su, UNIX-Befehl 177
sub, Perl-Subroutine 273
Subdomain (DNS) 607
Subnet Mask 1000
Subnet Mask, IP-Adresse 581
Subnetting 1000
Subnetting, IP-Netze teilen 582
Subroutinen
Parameter 274
Wertrückgabe 274
Subroutinen, Perl 273
SubSeven, Backdoor 977
substring(), Java-Methode 257
Subtraktion, Operator 235
Such-Algorithmen 284
Binäre Suche 285
Lineare Suche 284
Suchmaschinen
Anmeldung bei 761
HTML aufbereiten für 760
robots.txt-Datei 761
SUM, SQL-Funktion 379
Sun Microsystems
Java 251
Solaris, Betriebssystem 148
Sun SPARC, Prozessor 105
super, Java 261
Supernetting 1000
IP-Netze zusammenfassen 582
Superuser 159, 172
SUSE Linux
Installation 201
yast 204
Yast, Installationsprogramm 636
SUSE Linux-Distribution 171
SVG 1000
SVG (Scalable Vector Graphics) 686
Swap-Partition 162
Swapping (Speicher) 102
SWF, Dateiformat, aus Flash exportieren 947
Swing, Java 304
Ereignisbehandlung 304
Tabellen 304
Switch 562
switch/case-Fallentscheidung 241
default-Wert 241
Symbolic Links 166
Symbolische Konstante 251, 1000
Symmetrische Verschlüsselung 987
sys/types.h, C-Bibliothek 296
Syslog 189, 1001
System V 1001
System V Init 202
System V IPC 160
System V, UNIX 148
System.err, Java 303
System.out, Java 255
Systemanalyse 331
Systemaufruf 1001
Systemaufrufe 108, 153, 155
CreateProcess() 160
fork() 158
kill() 159

pause() 159
programmieren 295
UNIX 156
Win32 API 156
Systembefehle
UNIX 179
UNIX-Dateimanipulation 179
UNIX-Systemverwaltung 185
UNIX-Textmanipulation 183
Systemkonfiguration
MacOS X 211
Windows 223
Systemprogramme 155
Linux 169
Optionen 175
Parameter 175
UNIX 179
UNIX-Dateimanipulation 179
UNIX-Systemverwaltung 185
UNIX-Textmanipulation 183
Systemsteuerung, Windows 223
Systemvariable 1001
Systemvariablen
path 175
setzen (UNIX) 176
Systray, Windows 219

T

Tabelle
erzeugen, SQL 373
löschen, SQL 374
Tabellen, in HTML 742
Tags, HTML 720
Tags, XML 687
Taktfrequenz
der CPU 103
der Mainboard 103
Multiplikator 103
praktische Bedeutung 104
Tanenbaum, Andrew 152
tar-Datei 205
Task Scheduler 153, 1001
Taskleiste, Windows 218
Systray 219
Tastatur 134
Zeichensatzeinstellung 523
TCP 601, 1001
Drei-Wege-Handshake 603
Funktionsweise 601
im Vergleich zu UDP 601
Paket-Header 601
Ports 603
Urgent Data 604
Verbindungsaufbau 603
TCP/IP 576, 1001
Anwendungsprotokolle 615
ARP, Netzzugang 577
DHCP 595
Dienstprogramme 640
DNS 606
Domain Name System 606
FTP, Anwendungsprotokoll 617
HTTP, Anwendungsprotokoll 627
ICMP-Protokoll 601
IMAP, Anwendungsprotokoll 624
Loopback-Interface 580
Nameserver 606
Netzzugang 576
NNTP, Anwendungsprotokoll 625
POP3, Anwendungsprotokoll 623
Routing 589
Routing-Protokolle 592
SMTP, Anwendungsprotokoll 619
Telnet, Anwendungsprotokoll 616
Transportprotokolle 601
TCP/IP-Dienstprogramme 640
dig 643
netstat 642
nslookup 642
ping 640
traceroute 641
tracert, Windows 641
TCP/IP-Schichtenmodell 545
TCP-Client-Socket, mit IO 670
TCP-Client-Sockets 668
TCP-Header 601
TCP-Port, für Sockets 666
TCP-Ports 603
Well-Known Ports 604
TCP-Server-Socket, mit IO 671
TCP-Server-Sockets 669
tcsh (erweiterte C-Shell) 174
TDD → Test-driven Development
T-DSL → DSL
Technische Informatik 26
teilerfremd 277
Teilmenge 61
Teilnetzmaske, IP-Adresse 581
Telefongespräch als Schichtenmodell 547
Telefonleitung
Pulswahl 571
Telefonverbindung 538
Telnet 616, 1001
Terabyte 70
Term 54
TERM, Signal 159
Terminal 37, 147, 1001
unter grafischer Oberfläche 173
virtuelles 173
Terminator 1001
SCSI 121
Test, Software-Engineering 335
Code-Review 335
Lasttests 335
Unit-Tests 335, 348
Test-driven Development 351
Test-first-Verfahren 338, 351
Testgetriebene Entwicklung → Test-driven Development
TeX, Textsatzsystem 525
Text
Datei anzeigen, Windows-Konsole 222
Dateien vergleichen 185
Dateiinhalt anzeigen (UNIX) 183
Editoren, UNIX 192
Emacs, Editor 195

Manipulationsbefehle, UNIX 183
vi, Editor 193
Wörter zählen, UNIX 185
TEXT, SQL-Datentyp 376
text-align, CSS-Angabe 767
TextArea, AWT-Klasse 311
Textdatei
anzeigen, Windows-Konsole 222
Inhalt anzeigen (UNIX) 183
Textdateiformate, Vorteile 685
text-decoration, CSS-Angabe 766
Texteditor
Emacs 195
für XML verwenden 686
unter UNIX 192
vi 193
vim 193
TextField, AWT-Klasse 311
text-indent, CSS-Angabe 767
TFT 1001
TFT-Monitor 139
Theoretische Informatik 25
Thermosublimationsdrucker 141
Thermotransferdrucker 141
Thicknet Coaxial, Ethernet 561
Thinnet Coaxial, Ethernet 560
this, Java 258
Thompson, Ken 147
Thread 1001
Thread, Java-Klasse 300
Threads 152, 161
in Programmiersprachen 299
Java 262, 300
run() 300
Runnable-Interface 300
throws-Klausel, Java 263
TIFF, Bilddateiformat 531
time(), C-Funktion 250
time(), PHP-Funktion 839
TIME, SQL-Datentyp 375
time.t, C-Bibliothek 250
time_t, C-Datentyp 250
Timesharing 147, 1001
TIMESTAMP, SQL-Datentyp 375
Tintenstrahldrucker 140
Bubble-Technik 140
Piezo-Technik 140
TINYBLOB, SQL-Datentyp 376
TINYINT, SQL-Datentyp 375
TINYTEXT, SQL-Datentyp 376
Titel, HTML-Dokument 721
TLS → SSL
Token-Passing 1001
Token-Passing, Netzzugangsverfahren 563
Token-Ring-Netzwerk 563, 1001
Tomlinson, Ray 540
Tonkanäle, Audio 51
Tonwahlverfahren 571
Tonwertkorrektur, Photoshop 403
Top Level Domain 1001
Top Level Domain, Länder 608
top, CSS-Angabe 769
Top-Level-Domain (DNS) 607
Top-Level-Domain, Generic 608
Torvalds, Linus 149, 169
tr///, Perl-Operator 295
traceroute, TCP/IP-Dienstprogramm 641
Track-at-once 130, 1001
Transaktionen (RDBMS) 364
in MySQL 364
Transformation
Illustrator 418
InDesign 442
Photoshop 403
Transistor 36, 37, 74
TFT 139
Transistorrechner 36
Transparenz, Illustrator 416
Transport, OSI-Schicht 544
Treiber → Gerätetreiber
Triode 37
tripwire 985
Trojaner → Trojanische Pferde
Trojanische Pferde 976
Trolltech 199
Trommelscanner 135
Foto-Multiplier 136
true, Java 256
TrueType 1001
TrueType-Schrift 440
try, Java 255
try/catch-Blöcke, Java 255
TTL 1001
TTL, IP-Datagramm 592
Turing, Alan 84
Turing-Maschine 85, 1001
Band 85
Beispiele 85, 86
einfaches Beispiel 85
komplexeres Beispiel 86
Schreib-Lese-Kopf 85
Zeichenvorrat 85
Zustände 85
Turing-Test 84
Turing-Vollständigkeit 87, 1001
TV-Tunerkarte, für Digitalvideo-Aufnahmen verwenden 460
Tweening 1002
Director 481
Flash 932
Twisted-Pair-Kabel 561
Kategorien 561
type, Windows-Befehl 222
Typecasting, C 235
Typenraddrucker 140
TYPO3, CMS 808
Typographie 421
Absatz 430
Antiqua 426
Antiqua-Varianten 428
Barock-Antiqua 426
Buchstabenentwicklung 421
Dickte 423
Durchschuss 424
Expert-Zeichensatz 426
Festbreitenschrift 425
Fraktur 428
Französische Renaissance-Antiqua 426
Gebrochene Schriften 428
Gemeine 425

Gestaltung 433
Geviert 424
Großbuchstaben 424
Grundlinie 423
Handschriftliche Antiqua 428
Hurenkind 433
Kapitälchen 425
Kapitale 424
Kegelhöhe 422
Kerning 424
Klassizistische Antiqua 428
Kleinbuchstaben 425
kompress 424
kursiv 425
Laufweite 424
Ligaturen 425
Majuskeln 424
Majuskelziffern 426
Mediävalziffern 426
Mikrotypographie 429
Minuskeln 425
Minuskelziffern 426
Mittellänge 423
Oberlänge 423
Proprotionalschrift 425
Satzspiegel 430
Satzzeichen 429
Schreibschriften 428
Schriftfamilien 426
Schriftwahl 428
Schusterjunge 433
Serifenbetonte Linear-Antiqua 428
Serifenlose Linear-Antiqua 428
splendid 424
Unterlänge 423
Unterschneidung 424
Venezianische Renaissance-Antiqua 426
Versalhöhe 423
Versalien 424
Zahlen 430
Zeile 430
Zeilenabstand 424
Zeilenfall 432
Ziffern 426

U

Überladen 259
Übertakten 103
UCR, Under Color Removal 390
UCSD-Pascal 317
UDDI 1002
UDDI, Web Services 682
UDF 133
UDP 604, 1002
Anwendungsbeispiel 604
im Vergleich zu TCP 601
Paketheader 605
Ports 605
UDP-Header 605
UDP-Ports 605
UID (User-ID) 172
von Prozessen 159
Umbenennen
Dateien, MacOS X 211
Dateien, UNIX 180
Umfangsmanagement 328
Umfließen, Elemente durch Text 436
Umfließen, InDesign 445
Umgebungsvariablen
CLASSPATH 252
path 175
setzen (UNIX) 176
Setzen unter Windows 221
Umkehrschluss 56
UML 81, 339, 1002
Akteure 341
Aktivitätsdiagramme 344
Anwendungsfalldiagramme 341
ArgoUML, Tool 340
Diagrammtypen 339
Klassendiagramme 341
Praktischer Einsatz 340
Sequenzdiagramme 343
Version 2.0 339
umount, UNIX-Befehl 186
Umrechnung
Dezimal nach Dual 67
Dezimal nach Hexadezimal 68
Dual nach Dezimal 67
Dual nach Hexadezimal 69
Dual nach Oktal 68
Hexadezimal nach Dezimal 68
Hexadezimal nach Dual 69
Oktal nach Dual 68
Zahlensysteme 67
Unbedingter Sprung 106
Undefiniertheitsstelle (Funktion) 82
Und-Schaltung → AND-Schaltung
Und-Verknüpfung → AND-Verknüpfung
Ungleichheit 59
Ungleichheit, Operator 236
Ungleichung 54
Lösung 55
Unicode 521, 1002
BMP-Teilmenge 521
Tabelle wichtiger Teilzeichensätze 521
Unterstützung durch Software 522
UTF-8-Codierung 521
Unicode, Zeichensatz 52
Unified Modeling Language → UML
Unified Process 336
Aktivitäten 337
Anwendungsfälle 336
Artefakte 337
Phasen 337
Rollen 337
Vorgehen 337
UNIQUE, SQL-Schlüsselwort 377
Unit-Tests 335, 348
Grüner Balken 350
JUnit-Framework 349
Motivation 348
Roter Balken 350
Universal Disk Format → UDF
University of California, Berkeley 148, 193, 664
UNIX 46, 148
$0, Systemvariable 174
/etc/passwd-Datei 172

/etc/shadow, Datei 173
als Server einrichten 647
Apache-Installation 655
Arbeitsverzeichnis anzeigen 181
auf NFS-Freigaben zugreifen 648
auf Windows-Server zugreifen 650
bash 174
Benutzerrechte 166
Berkeley System Distribution (BSD) 148
Bourne-Shell 174
BSD 148
BSD-Startstkripte 204
cat-Befehl 183
cd-Befehl 181
chgrp-Befehl 183
Child-Prozess 159
chmod-Befehl 182
chown-Befehl 183
cp-Befehl 180
C-Shell 174
CUPS, Drucksystem 648
Daemon 189
Darwin 148
date-Befehl 187
Dateibefehle 179
Dateibesitzer wechseln 183
Dateien kopieren 180
Dateien löschen 181
Dateien umbenennen 180
Dateien verschieben 180
Datei-Endungen 180
Dateigruppe wechseln 183
Dateinamen 165, 179
Dateinamen-Platzhalter 179
Dateisysteme 164
Datum und Uhrzeit ändern 187
Datum und Uhrzeit formatieren 187
diff-Befehl 185
du-Befehl 186
Emacs, Texteditor 195
Escape-Sequenzen 185
exit-Befehl 177
fg-Befehl 175
finger, Dienstprogramm 172
fork(), Systemaufruf 158
fsck-Befehl 186
Gerätedateien 163, 164
GNOME 157, 198, 200
Grafische Benutzeroberfläche 198
grep-Befehl 184
groupadd-Befehl 188
Group-ID 159, 172
gunzip-Befehl 205
gzip-Komprimierung 205
Hard Links 166
HIER-Dokument 178
Home-Verzeichnis 165
HP UX 148
IBM AIX 148
ifconfig-Befehl 636
init-Prozess 158
inodes 166
Installation von Software 205
IP-Adresse zuweisen 637
KDE 157, 198, 199
kill(), Systemaufruf 159
kill-Befehl 190
Korn Shell 174
less-Befehl 183
Links (Dateisystem) 166
Linux 148, 149, 151, 169
Login 171
ls-Befehl 181
Mac OS X 148, 208
Macintosh im Netz bedienen 651
make-Befehl 205
man-Befehl 157
Minix 151
mkdir-Befehl 182
mkfs-Befehl 186
Modularität 148
more-Befehl 183
mount-Befehl 166, 185
mv-Befehl 180
MySQL-Installation 369
netatalk-Server 651
Neustart 190
NFS 647
Pager 183
Parent-Prozess 159
passwd-Befehl 188
Passwort ändern 188
patch-Befehl 185
path, Umgebungsvariable 175
pause(), Systemaufruf 159
Pfadangaben 165
PHP-Installation 811
Pipes 178
POSIX-Standard 148
Prozessmodell 158
Prozessverwaltung 189
ps-Befehl 159, 189
pstree-Befehl 189
pwd-Befehl 181
Reguläre Ausdrücke suchen 184
rm-Befehl 181
rmdir-Befehl 182
root, Benutzer 159, 172
route-Befehl 637
Runlevel 203
Samba-Server 649
Shell 157, 171
Shell-Ausgabeumleitung 177
Shell-Eingabeumleitung 178
Shell-Eingabevervollständigung 176
Shell-History 176
Shell-Skripte 190
shutdown-Befehl 190
Software installieren 205
startx-Befehl 199
su-Befehl 177
Sun Solaris 148
Swap-Partition 162
Symbolic Links 166
Syslog 189
System herunterfahren 190
System V 148
Systemaufrufe 156
Systemprogramme 179
tar-Dateien 205
Textbefehle 183
Textdatei anzeigen 183
Textdateien vergleichen 185

Texteditoren 192
Umgebungsvariablen setzen 176
umount-Befehl 186
useradd-Befehl 188
User-ID 159, 172
Verwaltungsbefehle 185
Verzeichnis anlegen 182
Verzeichnis löschen 182
Verzeichnis wechseln 181
Verzeichnisbaum 164
Verzeichnisbefehle 179
Verzeichnisinhalt anzeigen 181
vi, Editor 193
virtuelles Terminal 173
wc-Befehl 185
Window-Manager 157
Wörter zählen 185
X Window 157, 198
Zugriffsrechte 166
UNIX System V Init 202
unless, Perl-Fallentscheidung 273
unshift, Perl-Anweisung 268
unsigned, C-Datentyp 233
UNSIGNED, SQL-Feldoption 376
Unterlänge, Typographie 423
Unterprogramm, Aufruf durch CPU 106
Unterschneidung, Typographie 424
update(), AWT-Methode 309
UPDATE, SQL-Abfrage 381
URL 1002
Schema 629
URL, Web-Adresse 627
USB 122, 1002
Use Cases → Anwendungsfälle
use, Perl-Direktive 267
Usenet 625, 1002
User Level Security, Samba 650
User Level Security, Windows-Freigabeart 645
useradd, UNIX-Befehl 188
User-ID 172
von Prozessen 159
Users, MacOS X-Verzeichnis 165
usr, UNIX-Verzeichnis 164
UTF-8 1002
UTF-8, Unicode-Codierung 521

V

V.24-Schnittstelle 122
Validatoren, HTML 721
Validierung
HTML 721
var, UNIX-Verzeichnis 165
VARCHAR, SQL-Datentyp 376
Variablen
Arrays, Perl 267
automatische (lokale) 234
C 233
Datentypen, C 233
Deklaration, C 230, 232, 233
Deklaration, Java 256
deklarieren, Perl 267
Existenz testen, Perl 273
globale 234
Gültigkeitsbereich, C 234
Hashes, Perl 268
in Programmiersprachen 60
in Shell-Skripten 191
Lingo 498
lokale 234
mathematische 54
Perl 267
PHP 818
Schleifenzähler 242
Skalare, Perl 266, 267
statische 234
Substitution, Perl 266
Substitution, Shell-Skripte 192
Typecasting 235
Variablensubstituition, in Shell-Skripten 192
Variablensubstitution
in Perl-HIER-Dokumenten 272
Perl 266
VAX, Minicomputer-Serie von DEC 38
Vektorgrafik 50, 387, 1002
Illustrator 412
Vektorrechnung 80
Venezianische Renaissance-Antiqua 426
Vereinigungsmenge 64
Vererbung 47, 260, 1002
Interfaces, Java 261
Vergleichsoperationen 59
beim virtuellen Prozessor 90
Umkehrung 59
Vergleichsoperatoren 236
SQL 378
Verhaltensmuster 345
Verknüpfungen, logische 55
Versalhöhe, Schriftgröße 423
Verschieben
Dateien, MacOS X 211
Dateien, UNIX 180
Dateien, Windows 219
Verschlüsselung
→ Kryptografie
Verteilte Anwendung 674
Aufgaben 674
Frameworks im Vergleich 675
J2EE-Plattform 675
vertical-align, CSS 767
Verzeichnis
anlegen, UNIX 182
anlegen, Windows 222
Arbeitsverzeichnis anzeigen, UNIX 181
Inhalt anzeigen, UNIX 181
löschen, UNIX 182
löschen, Windows 222
wechseln, UNIX 181
wechseln, Windows 222
Verzeichnisdienst 1002
Active Directory 646
Verzeichnisse, verarbeiten, Perl 275
vi, Texteditor 193
Befehlsmodus 193
Dateibefehle 195
Editiermodus 193
kopieren 194

Löschen 194
Navigation 193
Suchfunktionen 194
Video
Analog-Schnittkarte 460
Dateiformate 534
digitalisieren 459
DV-Camcorder 459
DVD auslesen 460
in Flash importieren 945
Schnittsoftware 460
SMPTE-Zeitformat 462
Video-Dateiformate 534
AVI 535
MPEG 535
QuickTime 535
Video-DVDs
auf dem PC abspielen 133
Region-Code 133
Videoschnitt 459
A/B-Bearbeitung 462
Analog-Schnittkarte 460
Arbeitsablauf 463
Bluescreen 465
Filmdatei exportieren 466
Software 460
Titel erzeugen 465
TV-Karte 460
Überblendungen 465
Vierfarbdruck 389
vim, Texteditor 193
Viren 44, 972
Antivirenprogramme 974
Aufbau 972
Bootsektor-Viren 973
Dateiviren 973
Makroviren 973
polymorphe 973
Schutzmaßnahmen 974
Stealth-Viren 973
Virtual Private Network → VPN
Virtuelle Hosts 662
Virtuelle Maschine, Java 251
Virtueller Prozessor 87
Arbeitsspeicher 88
Aufbau 88
Befehle 89
Beispielprogramme 92
Flags 90
Rechenbefehle 89
Register 88
Sprungbefehle 90
Vergleichsoperationen 90
virtueller Prozessor, Stack 91
Virtueller Speicher 102, 161
Visual Studio, IDE 317
Vlissides, John 345
VLSM 1002
VLSM, variable IP-Teilnetze 586
VMS, Betriebssystem 150
void, Datentyp
Zeiger auf 287
void, Funktionsdatentyp, C 243
Volladdierer (Schaltung) 77
Vollbildfenster, JavaScript 894
Volltextdatenbank 356
Volltonfarben 390
von Neumann, John 36, 96
Von-Neumann-Rechner 87, 96, 1002
VPN 989

W

-w, Perl-Modifikator 265
W3C, DOM 716
Wahre Aussagen 54
Wahrheitstabelle 57
AND-Verknüpfung 57
NAND-Verknüpfung 75
NOR-Verknüpfung 75
OR-Verknüpfung 58
XOR-Verknüpfung 59
Wall, Larry 264
WAN 551, 1002
Technische Lösungen 551
Warnock, John 40
Warnungen, in Perl aktivieren 265
Wasserfallmodell 327
WAV, Audio-Dateiformat 533
Wavetable-Synthese (MIDI) 143
wc, UNIX-Befehl 185
wchar_t, C-Datentyp 234
Wearable Computers 42
Web Services 681, 1002
Beschreibungsdatei 682
Clients 683
Definition 681
registrieren 682
SOAP 682
Sprachen 682
UDDI 682
WSDL 682
Webanwendungen
CGI 807
Grundprinzip 807
Sicherheitsprobleme 984
Web-Browser
Funktionsunterschiede 775
Internet Explorer 775
KDE Konqueror 199
Mozilla 775
Netscape 775
Opera 775
Webdesign 771
Bildauflösung 778
Bildbearbeitung 776
Bildgröße 778
Bildschrimäuflösung 772
Corporate Design berücksichtigen 774
Farbkontraste 774
Farbpalette 773
Farbtiefe 773
Grundregeln 772
Layout-Regeln 773
mit Dreamweaver 790
Test in verschiedenen Browsern 775
Web-Palette verwenden 773
WebLogic, Java Application Server 675
Weblogs 809
Blogging-Communities 809
Weblog-Software 809
Web-Palette 773
Web-Safe Colors 734
Webserver 556
Apache 652
Programmierung 807
Websichere Farben 734

Websites
Design 771
Erstellen in Dreamweaver 790
in Dreamweaver verwalten 798
robots.txt-Datei 761
Test im Browser 775
Zugang für Sehbehinderte 772
Wechseldatenträger 128
CD-ROM 129
Diskettenlaufwerk 128
DVD 132
Jaz-Laufwerk 128
LS-120 128
Zip-Laufwerk 128
Weiche Auswahlkante, Photoshop 400
Weiß, Mischung aller Lichtfarben 388
Weizenbaum, Joseph 84
Well-Known Ports, Tabelle einiger wichtiger 605
wertdiskret 51
Wertrückgabe, Perl-Subroutinen 274
Wertuweisung, Operator 237
Wertzuweisung, C 232
What You See Is What You Get → WYSIWYG
WHERE-Klausel, SQL 378
while()-Schleife 242
while, Befehl, in Shell-Skripten 191
White Book 1003
White Book (Video-CD) 130
Whitespace
in C-Programmen 232
in RegExp 292
Wide Area Network 551
Widerspruch, logischer 82
Widgets 303
Wikipedia 809
Wikis 809
Wildcards
in UNIX-Dateinamen 179
Windows-Dateinamen 221
Win32 1003
Umgang mit 16-Bit-Anwendungen 215
Win32 API 156, 1003
WindowListener, AWT-Klasse 314
Window-Manager 198
CDE 198
fvwm2 198
Windows 150, 213
16-Bit-Versionen 213
2000 151, 214
2000 Server 214
95 151, 214
98 151, 214
Active Directory 646
als Server einrichten 644
Anzeige-Konfiguration 223
Apache-Installation 656
attrib-Befehl 222
auf Netzwerkrechner zugreifen 645
Ausgabeumleitung 221
Batch-Datei 223
Benutzer-An- und Abmeldung 219
Benutzerverwaltung 224
cd-Befehl 222
copy-Befehl 222
CreateProcess(), Systemaufruf 160
CygWin 228
Dateiattribute 168
Dateiattribute ändern 222
Dateien kopieren 219
Dateien löschen, Konsole 222
Dateien umbenennen 222
Dateien verschieben 219, 222
Dateinamen 168
Dateinamen-Platzhalter 221
Dateisysteme 167, 215
Datenträgerverwaltung 225
del-Befehl 222
Desktop-Konfiguration 223
Dienste 225
dir-Befehl 222
DOSKEY 221
Drucker freigeben 645
durch UNIX-Server bedienen 649
Eingabeaufforderung 221
Eingabeumleitung 221
Entwicklung 213
Ereignisanzeige 225
Explorer 219
Fenster 218
Freigabeart 645
Hardware-Konfiguration 223, 224
help-Befehl 157
herunterfahren 219
Home-Verzeichnis 168
IP-Adresse zuweisen 639
Konsole 221
Konsole starten 221
Konsolenbefehle 222
Laufwerksbuchstaben 167
MDI-Anwendung 218
Me 151, 214
Microsoft Management Console (MMC) 225
mkdir-Befehl 222
MMC (Microsoft Management Console) 225
move-Befehl 222
MySQL-Installation 370
Netzwerkdrucker nutzen 645
Netzwerkkonfiguration 639
Netzwerktreiber 639
Netzwerkverbindungen einrichten 639
neu starten 219
NT 150, 214
NT Server 214
NT-Familie 214
Oberfläche 216
Ordner freigeben 645
Ordner-Ansichten 220
path, Umgebungsvariable 221
Pfadangaben 167
PHP-Installation 813
Pipes 221
Prompt 221
Prozessmodell 160

regedit, Dienstprogramm 225
Registry 225
rename-Befehl 222
rmdir-Befehl 222
SDI-Anwendungen 218
Server 2003 214
Serverpakete 647
Serversysteme 646
Server-Versionen 214
Share Level Security 645
Software installieren 223
StartmenüStartmenü, Windows 218
Systemkonfiguration 223
Systemsteuerung 223
Systray 219
Taskleiste 218
Textdatei anzeigen, Konsole 222
type-Befehl 222
Umgebungsvariablen setzen 221
User Level Security 645
Versionsübersicht 214
Verzeichnis anlegen 222
Verzeichnis löschen 222
Verzeichnis wechseln 222
Verzeichnisinhalt, Konsole 222
Win32 API 156
XP 151, 214
Windows 2000 151, 214
Server 214
Shell 221
Windows 2000 Server 646
Windows 95 151, 214
Oberfläche 217
Windows 98 151, 214
Oberfläche 217
Windows Me 151, 214
Windows NT 150, 214
Server 214
Shell 221
Windows Server 2003 214, 646
als DHCP-Server 646
als DNS-Server 646
als Webserver 646
Server-Rollen 646
Varianten 646
Windows XP 151, 214
Oberfläche 217
Shell 221
WinGate, Backdoor 977
Winkelgeschwindigkeit, konstante 128
WinTel-PC 40
Wireless LAN 564, 565
Access Point 567
Basic Service Set (BSS) 566
Extended Servive Set (ESS) 567
Frequency Hopping 566
Frequenzbereiche 566
Hardware 566
Sicherheit 567
Übertragungstechnik 565
Wirelesse LAN, CSMA/CA-Verfahren 566
Wirth, Niklaus 46
Wirtschaftsinformatik 32
WLAN → Wireless LAN
Wohlgeformtheit 1003
Wohlgeformtheit, XML-Dokumente 693
Word, 16 Bit 70
World Wide Web 626
Geschichte 541
HTTP-Protokoll 627
Komponenten 542
URL 627
Worst Case (Komplexität) 82
Wortbreite 39, 69, 1003
der CPU 102
verschiedener CPUs 103
Wörter zählen (Textdateien) 185
Wozniak, Steve 40, 149
WSDL 1003
WSDL, Web Services 682
Würmer 973
Schutzmaßnahmen 974
WYSIWYG 39, 1003

X

X Window 157, 198, 1003
Konfiguration 198
SaX, SUSE-Konfigurationsprogramm 199
starten 199
XConfiguratoe, RedHat-Konfigurationsprogramm 199
X11R6 198
x86-Assembler, Beispiele 107
Xalan, XSLT-Prozessor 710
XAMPP 652
XConfigurator (X-Server-Konfigurationsprogramm) 199
Xerces, XML-Parser 712
XEROX PARC 39
Xerox PARC 150
XFree86 198
XHTML 719, 1003
Besonderheiten 722
Dokumentstruktur 721
XML 685, 1003
Attributdeklaration (DTD) 702
Attribute 687, 690
Attribute, falsche Verwendung 691
Attribute, Verwendung 691
Beispieldokument 687
CDATA-Blöcke 692, 693
CSS verwenden 709
Dateiendung 686
Datenbank 356
DocBook 686
DOCTYPE-Angabe 696
Dokument parsen, SAX 713
Dokumenteingabe 686
Dokument-Fragmente 694
Dokumentstruktur 687
DTD 696
DTD definieren 696
DTD, Attribute deklarieren 702
DTD, Elemente deklarieren 698
DTD, Entities deklarieren 704
EJB-Deployment-Deskriptor 679

Elementdeklaration (DTD) 698
Elemente (Tags) 688
Entitites deklarieren (DTD) 704
Entity-Referenzen 692, 704
Entity-Referenzen vermeiden 693
Entity-Referenzen, numerisch 692
Hierarchie 689
in Datenbank schreiben, Beispielprogramm 715
leere Tags 690
leere Tags, Kurzfassung 690
mehrere Namensräume 706
MIME-Types 686
Namensräume 705
Namensräume, mehrere im Dokument 706
Parser 712
Parser, Xerces 712
PCDATA 692
Processing Instructions (PI) 688
Programmierung 712
Programmierung, SAX 713
PUBLIC-ID 696
SAX 713
Schema 706
Schema-Beispiel 707
SGML-Erbe 685
Sonderzeichen 691
spezielle Editoren 686
Standalone-Dokument 688
Standard-Namensraum 705
Steueranweisungen 688
SVG 686
SYSTEM-ID 696
Tag-Namen 689
Tags 687
Tag-Verschachtelung 689
Universalität 686
Verschachtelte Tags 689
verschachtelte Tags 695
Verschachtelungsfehler 695
Web Services 681
wichtige Dokumentformate 686
Wohlgeformtheit 693
Wurzelelement 694
Xerces, Parser 712
XHTML 686
xmlns-Angabe 706
xml-Steueranweisung 688
XSL 709
XSL-FO 709
XSLT 709
Zeichensätze 688
XML Schema 706, 1003
Attribute deklarieren 708
Elemente deklarieren 707
Verschachtelte Elemente 708
XML-Datenbank 356
XML-Editoren 686
XMLReader, Java-Interface 713
XOR-Schaltung 76
XOR-Verknüpfung 58
XP → Extreme Programming
X-Server 157, 198
Konfiguration 198
über das Netzwerk betreiben 616
XSL 709
XSL Formatting Objects 709
XSL-FO 709, 1003
XSLT 709, 1003
Prozessor 710
Wichtige Elemente 711
Wurzelelement 711
Xalan, Prozessor 710
xsl
for-each, Tag 712
template, Tag 712
value-of, Tag 712
XSLT-Prozessor 710
XSS → Cross-Site-Scripting

Y

YaST, SUSE-Installationsprogramm 636
YaST, SUSE-Konfigurationstool 204
YEAR, SQL-Datentyp 375
Yellow Book (CD-ROM) 129

Z

Z3, erster funktionierender Computer 35
Z80, Prozessor 103
Zahlen, Typographie 430
Zahlenmengen
Ganze Zahlen 62
Imaginäre Zahlen 63
Irreelle Zahlen 63
Komplexe Zahlen 63
Natürliche Zahlen 62
Rationale Zahlen 62
Reelle Zahlen 62
Zahlensysteme 64
Dezimalsystem 65
Dualsystem 65
Hexadezimalsystem 66
Oktalsystem 66
Römische Zahlen 65
Schreibweise 69
Umrechnung 67
Zauberstab, Photoshop 400
Zeichen
alternative in RegExp 291
aus Strings lesen, Java 257
ausschließen in RegExp 291
beliebig viele, in RegExp 291
Darstellung, C 233
eines oder mehr, in RegExp 291
Eingabe, C 248
genaue Anzahl, in RegExp 292
Gruppen in RegExp 291
optionale, in RegExp 291
Wortbestandteile in RegExp 292
Zeichengeräte (char devices) 154
Zeichenkette → String
Zeichen-Literale 235
Zeichensatz, in HTML angeben 723
Zeichensätze 51, 518
ANSI 520

ASCII 51
Codepage 520
Eingabe chinesischer Zeichen 524
Unicode 52, 521
Zeichenstift, InDesign 441
Zeichenvorrat
der Turing-Maschine 85
Zeiger 244, 1003
auf beliebigen Datentyp 287
auf nichts (NULL) 250
fehlende in Java 257
für Call by Reference 245
Zeile, Typographie 430
Zeilenabstand 424
optischer 424
Zeilenfall 432
Zeilenfrequenz (Monitor) 138
Zeilenumbruch
auf verschiedenen Plattformen 516
HTML 722
konvertieren 517
Perl-Programme 517
Zeilenumbrüche, entfernen, Perl 266
zeitdiskret 51
Zeitmanagement 328
Netzplan 329
Zentraleinheit 98
alte Bedeutung 99
ZEROFILL, SQL-Feldoption 376
Ziffern
in RegExp 292
Majuskelziffern 426
Mediävalziffern 426
Minuskelziffern 426
Typographie 426
z-index, CSS-Angabe 769
Zip-Laufwerk 128
Zugriffsrechte 166
ändern, UNIX 182
Zuordnungseinheiten (Dateisystem) 163
Zuse, Konrad 35
Zustände, der Turing-Maschine 85
Zweierkomplement 71
ZX Spectrum 149
ZX81 40, 149